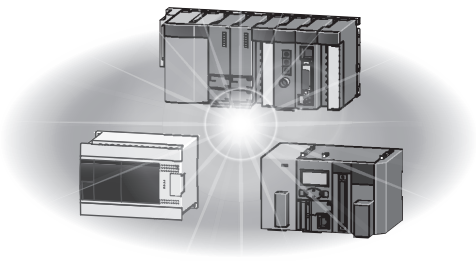
MITSUBISHI ELECTRIC

Mitsubishi Programmable Controller

MELSEC Q series MELSEC L series

MELSEC-Q/L
Programming Manual (MELSAP-L)



• SAFETY PRECAUTIONS •

(Read these precautions before using this product.)

Before using MELSEC-Q or -L series programmable controllers, please read the manuals included with each product and the relevant manuals introduced in those manuals carefully, and pay full attention to safety to handle the product correctly.

Make sure that the end users read the manuals included with each product, and keep the manuals in a safe place for future reference.

A - 1

• CONDITIONS OF USE FOR THE PRODUCT •

(1) Mitsubishi programmable controller ("the PRODUCT") shall be used in conditions;
i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.
MITSUBISHI SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI'S USER, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
("Prohibited Application")
Prohibited Applications include, but not limited to, the use of the PRODUCT in;
- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

Notwithstanding the above, restrictions Mitsubishi may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTs are required. For details, please contact the Mitsubishi representative in your region.

REVISIONS

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	Revision
Feb., 2000	SH (NA) 080076-A	First edition
May, 2001	SH (NA) 080076-B	Partial correction Chapter 1, Section 3.1, 5.1.1, 5.2.4, Appendix 1.2 Deletion Appendix 2
Apr., 2002	SH (NA) 080076-C	Partial correction Chapter 1, 2, Section 3.1, 5.1, 5.1.2, 5.2.4, Appendix 1.2
Mar., 2003	SH (NA) 080076-D	Descriptions on use of MELSAP-L with the Basic model QCPU whose serial number (first five digits) is "04122" or later have been added. Overall reexamination
Jun., 2004	SH (NA) 080076-E	Descriptions on the Redundant CPU have been added. Partial correction ABOUT MANUALS, Chapter 1, 2, Section 3.12, 3.2.2, 3.3, 4.2, 4.2.8, 4.3.3, 4.3.5, 4.4, 4.4.1 to 4.4.11, 4.5, 4.7, 5.2, 5.2.1, 5.2.2, 5.3.1, Appendix 1.1, 1.2, 2
May, 2005	SH (NA) 080076-F	Partial correction ABOUT MANUALS, Section 4.4.2, 4.4.9, 5.2.2, 6.6, Appendix 1.1 Addition Section 3.3.2, 4.8, 4.8.1, 4.8.2 Section number change Section 3.3 → Section 3.3.1
Apr., 2007	SH (NA) 080076-G	New models of the Universal model QCPU have been added. Model addition Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU Partial correction ABOUT MANUALS, GENERIC TERMS, Chapter 1, 2, Section 3.1.2, 3.2.2, 3.3.1, 4.2, 4.4.1 to 4.4.11, 4.5 to 4.7, 5.2, 5.2.1, 5.2.2, 5.3.1, 6.6, Appendix 1.1, 1.2, 2, 3
Dec., 2007	SH (NA) 080076-H	Partial correction Section 6.3.2
Mar., 2008	SH (NA) 080076-I	New models of the Universal model QCPU have been added. Model addition Q13UDHCPU, Q26UDHCPU Partial correction GENERIC TERMS, CHAPTER 2, SECTION 3.1.2, 3.2.2, 3.3.1, 4.2, 4.2.8, 4.3.3, 4.4.5, 5.2.2

A - 3

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	Revision
May, 2008	SH (NA) 080076-J	New models of the Universal model QCPU and Process CPU have been added. Model addition Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q13UDEHCPU, Q26UDEHCPU Q02PHCPU, Q06PHCPU Partial correction GENERIC TERMS, Chapter 2, Section 3.1.2, 3.3.1, 4.2, 4.2.8, 4.3.3, 4.7.1, 5.2.2
Dec., 2008	SH (NA) 080076-K	New models of the Universal model QCPU have been added. Model addition Q00UJCPU, Q00UCPU, Q01UCPU, Q10UDHCPU, Q20UDHCPU, Q10UDEHCPU, Q20UDEHCPU Partial correction ABOUT MANUALS, GENERIC TERMS, Section 1.2, Chapter 2, Section 3.1.2, 3.2.2, 3.3.1, 3.4, 4.2, 4.2.8, 4.3.3, 5.2.2, Appendix 1
Jul., 2009	SH (NA) 080076-L	The serial number (first five digits) of the Universal model QCPU has been upgraded to "11043". Partial correction Section 4.1, 4.7.1, 6.1.1
Jan., 2010	SH (NA) 080076-M	Descriptions on MELSEC-L series modules have been added. Partial correction ABOUT MANUALS, GENERIC TERMS, Chapter 1, Section 1.2, Chapter 2, Section 3.1.2, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 4.2, 4.2.2, 4.2.7, 4.2.8, 4.2.9, 4.2.10, 4.2.11, 4.3.3, 4.3.5, 4.4, 4.4.1 to 4.4.11, 4.5, 4.5.3, 4.6, 4.7, 4.7.3, 4.7.4, 4.7.5, 4.7.6, 4.8.1, 4.8.2, 5.2, 5.2.1, 5.2.2, 5.2.3, 5.3.1, 6.1, 6.1.1, 6.3.1, 6.3.2, 6.4.3, 6.6, Appendix 1.1, 1.2, 2 Addition CONDITIONS OF USE FOR THE PRODUCT, Section 3.2.4
Apr., 2010	SH (NA) 080076-N	New models of the Universal model QCPU have been added. Model addition Q50UDEHCPU, Q100UDEHCPU Partial correction GENERIC TERMS, Chapter 2, Section 3.1.2, 3.3.1, 4.2, 4.2.8, 4.3.3, 5.2.2
Dec., 2010	SH (NA) 080076-O	Partial correction Section 4.3.4
Jul., 2011	SH (NA) 080076-P	New models of the LCPU have been added. Model addition L02CPU-P, L26CPU-PBT Partial correction GENERIC TERMS, Section 1.2, Chapter 2, Section 3.1.2, 3.2.4, 3.3.1, 4.2, 4.2.8, 4.2.9, 4.3.3, 5.2.2
Nov., 2011	SH (NA) 080076-Q	The serial number (first five digits) of the Universal model QCPU has been upgraded to "13102". Partial correction SAFETY PRECAUTIONS, ABOUT MANUALS, Chapter 4, Section 4.4, 4.4.1, 4.4.8, 4.4.9, 4.4.10, 4.5, 6.1, Appendix 1, 2

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	Revision
Feb., 2013	SH (NA) 080076-R	New models of the Universal model QCPU and LCPU have been added. Model addition Q03UDVCPU, Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, Q26UDVCPU, L02SCPU, L06CPU, L26CPU Partial correction GENERIC TERMS, Section 1.2, Chapter 2, Section 3.1.2, 3.2.4, 3.3.1, 4.2, 4.2.8, 4.2.9, 4.3.3, 5.2.2
Jun., 2013	SH (NA) 080076-S	New models of the LCPU have been added. Model addition L02SCPU-P, L06CPU-P, L26CPU-P Partial correction GENERIC TERMS, Section 1.2, Chapter 2, Section 3.1.2, 3.2.4, 3.3.1, 4.2, 4.2.8, 4.2.9, 4.3.3, 5.2.2
Jun., 2014	SH (NA) 080076-T	The serial number (first five digits) of the LCPU has been upgraded to "16042". Partial correction Section 3.2.4

Japanese Manual Version SH-080072-V

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2000 MITSUBISHI ELECTRIC CORPORATION

INTRODUCTION

Thank you for purchasing the Mitsubishi MELSEC-Q/L series programmable controllers.
Before using the product, please read this manual carefully and develop familiarity with the functions and performance of the MELSEC-Q/L series programmable controllers to handle the product correctly.
Please make sure that the end users read this manual.

CONTENTS

SAFETY PRECAUTIONS....A- 1
CONDITIONS OF USE FOR THE PRODUCT....A- 2
REVISIONS....A- 3
INTRODUCTION....A- 6
CONTENTS....A- 6
ABOUT MANUALS....A- 9
GENERIC TERMS....A- 10

1. GENERAL DESCRIPTION 1- 1 to 1-10

1.1 SFC Program....1- 3
1.2 SFC (MELSAP-L) Features....1- 4

2. SYSTEM CONFIGURATION 2- 1 to 2- 2

3. SPECIFICATIONS 3- 1 to 3-26

3.1 Performance Specifications Related to SFC Programs....3- 1
3.1.1 When the Basic model QCPU is used....3- 1
3.1.2 When the High Performance model QCPU, Process CPU, Redundant CPU, Universal model CPU, or LCPU is used....3- 3
3.2 Device List....3- 8
3.2.1 Device list of Basic model QCPU....3- 8
3.2.2 Device list of High Performance model QCPU, Process CPU, and Redundant CPU....3-10
3.2.3 Device list of Universal model QCPU....3-12
3.2.4 Device list of LCPU....3-15
3.3 Processing Time....3-17
3.3.1 Processing time for SFC program....3-17
3.3.2 Processing time for S(P).SFCSCOMR instruction and S(P).SFCTCOMR instruction....3-22
3.4 Calculating the SFC Program Capacity....3-24

4. SFC PROGRAM CONFIGURATION 4- 1 to 4-114

4.1 List of SFC Diagram Symbols....4- 2
4.2 Steps....4- 5
4.2.1 Step □ (without step attribute)....4- 6
4.2.2 Initial step □....4- 9
4.2.3 Dummy step ☒....4-10
4.2.4 Coil HOLD step SC....4-10
4.2.5 Operation HOLD step (without transition check) SE....4-12
4.2.6 Operation HOLD step (with transition check) ST....4-14

4.2.7 Reset step [R] 4-16
4.2.8 Block START step (with END check) 4-17
4.2.9 Block START step (without END check) 4-19
4.2.10 End step 4-21
4.2.11 Instructions that cannot be used with operation outputs 4-23
4.3 Transition 4-24
4.3.1 Serial transition 4-25
4.3.2 Selection transition 4-27
4.3.3 Parallel transition 4-30
4.3.4 Jump transition 4-34
4.3.5 Precautions for creating operation output (step)/transition condition programs 4-35
4.4 Controlling SFC Programs by Instructions (SFC Control Instructions) 4-39
4.4.1 Step operation status check instructions (a, b, &a, &b, la, lb) 4-43
4.4.2 Forced transition check instruction (a, b, &a, &b, la, lb) 4-46
4.4.3 Block operation status check instruction (a, b, &a, &b, la, lb) 4-48
4.4.4 Active step batch readout instructions (MOV, DMOV) 4-50
4.4.5 Active step batch readout (BMOV) 4-53
4.4.6 Block START & END instructions (s, r) 4-57
4.4.7 Block STOP and RESTART instructions (PAUSE, RSTART) 4-58
4.4.8 Step START and END instructions (s, r) 4-61
4.4.9 Forced transition EXECUTE & CANCEL instructions (s, r) 4-65
4.4.10 Active step change instruction (SCHG) 4-67
4.4.11 Block switching instruction (BRSET) 4-68
4.5 SFC Information Devices 4-70
4.5.1 Block START/END bit 4-72
4.5.2 Step transition bit 4-74
4.5.3 Block STOP/RESTART bit 4-76
4.5.4 Block STOP mode bit 4-79
4.5.5 Continuous transition bit 4-82
4.5.6 "Number of active steps" register 4-84
4.6 Step Transition Watchdog Timer 4-85
4.7 SFC Operation Mode Setting 4-87
4.7.1 SFC program start mode 4-88
4.7.2 Block 0 START condition 4-90
4.7.3 Output mode at block STOP 4-91
4.7.4 Periodic execution block setting 4-92
4.7.5 Operation mode at double block START 4-93
4.7.6 Operation mode at transition to active step (double step START) 4-94
4.8 SFC Comment Readout Instruction 4-97
4.8.1 SFC comment readout instruction (S(P). SFCSCOMR) 4-98
4.8.2 SFC transition comment readout instruction (S(P). SFCTCOMR) 4-106

5. SFC PROGRAM PROCESSING SEQUENCE 5- 1 to 5-14

5.1 Whole Program Processing of Basic Model QCPU 5- 1
5.1.1 Whole program processing sequence 5- 1
5.2 Whole Program Processing of High Performance Model QCPU, Process CPU, Redundant CPU, Universal Model QCPU, and LCPU 5- 2
5.2.1 Whole program processing sequence 5- 2

5.2.2 Execution type designation by instructions 5- 4
5.2.3 SFC program for program execution management 5- 6
5.3 SFC Program Processing Sequence 5- 8
5.3.1 SFC program execution 5- 8
5.3.2 Block execution sequence 5-10
5.3.3 Step execution sequence 5-11
5.3.4 Continuous transition ON/OFF operation 5-12

6. SFC PROGRAM EXECUTION 6- 1 to 6-25

6.1 SFC Program START And STOP 6- 1
6.1.1 SFC program resumptive START procedure 6- 2
6.2 Block START and END 6- 4
6.2.1 Block START methods 6- 4
6.2.2 Block END methods 6- 5
6.3 Block Temporary Stop and Restart Methods 6- 6
6.3.1 Block STOP methods 6- 6
6.3.2 Restarting a stopped block 6- 9
6.4 Step START (Activate) and END (Deactivate) Methods 6-11
6.4.1 Step START (activate) methods 6-11
6.4.2 Step END (deactivate) methods 6-12
6.4.3 Changing an active step status (Not available for Basic model QCPU, Universal model QCPU, and LCPU) 6-13
6.5 Operation Methods for Continuous Transition 6-14
6.6 Operation at Program Change 6-15
6.6.1 Operation at program change made by write to PLC 6-17
6.6.2 Program change by online change 6-18
6.6.3 Online change (inactive block) 6-19

APPENDICES APP- 1 to APP-11

APPENDIX 1 Special Relay and Special Register List ..... APP- 1
APPENDIX 1.1 Special Relay (SM) List ..... APP- 2
APPENDIX 1.2 Special Register (SD) List ..... APP- 6
APPENDIX 2 Restrictions on Basic Model QCPU, Universal Model QCPU, and LCPU and Alternative Methods ..... APP- 7
APPENDIX 2.1 Step Transition Watchdog Timer Replacement Method ..... APP- 8
APPENDIX 2.2 Periodic Execution Block Replacement Method ..... APP- 9
APPENDIX 2.3 Forced Transition Bit (TRn) Replacement Method APP-10
APPENDIX 2.4 Active Step Change Instruction (SCHG) Replacement Method APP-11

ABOUT MANUALS

The manuals related to this product are listed below.
Order each manual as needed, referring to the following lists.

Relevant manuals

Manual name	Manual number (model code)
GX Developer Version 8 Operating Manual (SFC) Describes how to create SFC programs using the software package for creating SFC programs. (Sold separately)	SH-080374E (13JU42)
GX Works2 Version1 Operating Manual (Common) Describes system configuration, parameter settings, and online operations (common to Simple project and Structured project) of GX Works2. (Sold separately)	SH-080779ENG (13JU63)
QnUCPU User's Manual (Function Explanation, Programming Fundamentals) Describes the functions, programming procedures, devices, etc. necessary to create programs using the CPU modules. (Sold separately)	SH-080807ENG (13JZ27)
Qn(H)/QnPH/QnPRHCPU User's Manual (Function Explanation, Program Fundamentals) Describes the functions, programming procedures, devices, etc. necessary to create programs using the CPU modules. (Sold separately)	SH-080808ENG (13JZ28)
MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals) Describes the functions required for programming, programming methods, and devices. (Sold separately)	SH-080889ENG (13JZ35)
MELSEC-Q/L Programming Manual (Common instruction) Describes how to use sequence instructions, basic instructions, and application instructions. (Sold separately)	SH-080809ENG (13JW10)

A - 9

GENERIC TERMS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term	Description
QCPU	A generic term for the Basic model QCPU, High Performance model QCPU, Process CPU, Redundant CPU, and Universal model QCPU
QnCPU	A generic term for the Q02CPU
QnHCPU	A generic term for the Q02HCPU, Q06HCPU, Q12HCPU, and Q25HCPU
QnPHCPU	A generic term for the Q02PHCPU, Q06PHCPU, Q12PHCPU, and Q25PHCPU
QnPRHCPU	A generic term for the Q12PRHCPU and Q25PRHCPU
LCPU	A generic term for the L02SCPU, L02SCPU-P, L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, and L26CPU-PBT
Basic model QCPU	A generic term for the Q00JCPU, Q00CPU, and Q01CPU
Basic	
High Performance model QCPU	A generic term for the Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, and Q25HCPU
High Performance	
Process CPU	A generic term for the Q02PHCPU, Q06PHCPU, Q12PHCPU, and Q25PHCPU
Redundant CPU	A generic term for the Q12PRHCPU and Q25PRHCPU
Universal model QCPU	A generic term for the Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDVCPU, Q03UDECPU, Q04UDHCPU, Q04UDVCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDVCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDVCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDVCPU, Q26UDEHCPU, Q50UDEHCPU, and Q100UDEHCPU
Universal	
High-speed Universal model QCPU	A generic term for the Q03UDVCPU, Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, and Q26UDVCPU
Programming tool	A generic term for GX Developer and GX Works2

MEMO

A - 11

1. GENERAL DESCRIPTION

SFC, an abbreviation for "Sequential Function Chart", is a control specification description format in which a sequence of control operations is split into a series of steps to enable a clear expression of the program execution sequence and execution conditions.

This manual describes the specifications, functions, instructions, programming procedures, etc. used to perform programming with an SFC program using MELSAP-L.

MELSAP-L can be used with the following CPU modules.
- Basic model QCPU whose serial number (first five digits) is 04122 or later
- High Performance model QCPU
- Redundant CPU
- Process CPU
- LCPU
- Universal model QCPU

MELSAP-L conforms to the IEC Standard for SFC.
In this manual, MELSAP-L is referred to as SFC (program, diagram).

POINT

(1) The following functions cannot be executed if a parameter that sets the "high speed interrupt cyclic interval" is loaded into a High Performance model QCPU of which the first 5 digits of the serial number are "04012" or later.
- Step transition watch dog timer (see Section 4.6)
- Periodic execution block setting (see Section 4.7.4)

(2) The Qn(H)CPU-A (A mode) cannot use MELSAP-L explained in this manual.

1 - 1

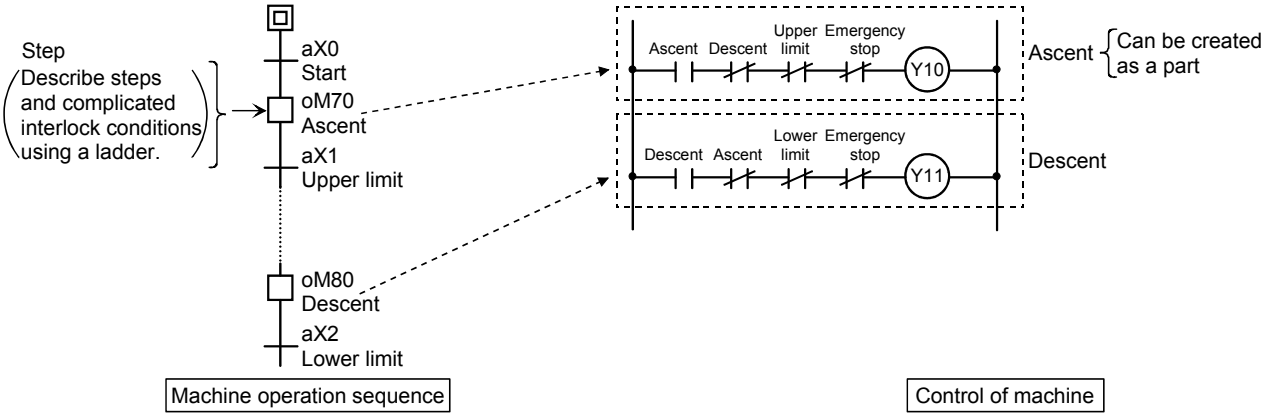
(1) When created with MELSAP-L and ladders

(a) MELSAP-L side

The flow of operation is easy to understand by creating the SFC program related to the interlock conditions.

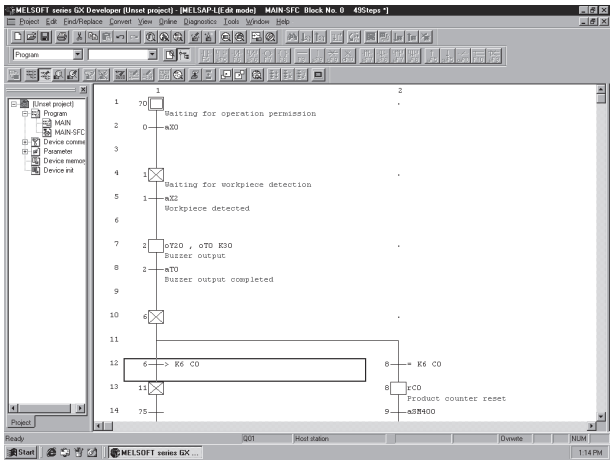
(b) Sequence programs side

The area can be developed into a product by creating interlock conditions irrelevant to the flow of operation.



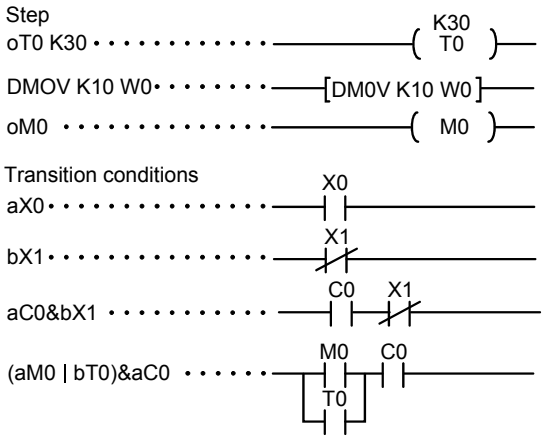
(2) Description format with MELSAP-L

MELSAP-L display screen



The description format in the step and transition conditions with MELSAP-L is shown b.

(Example)

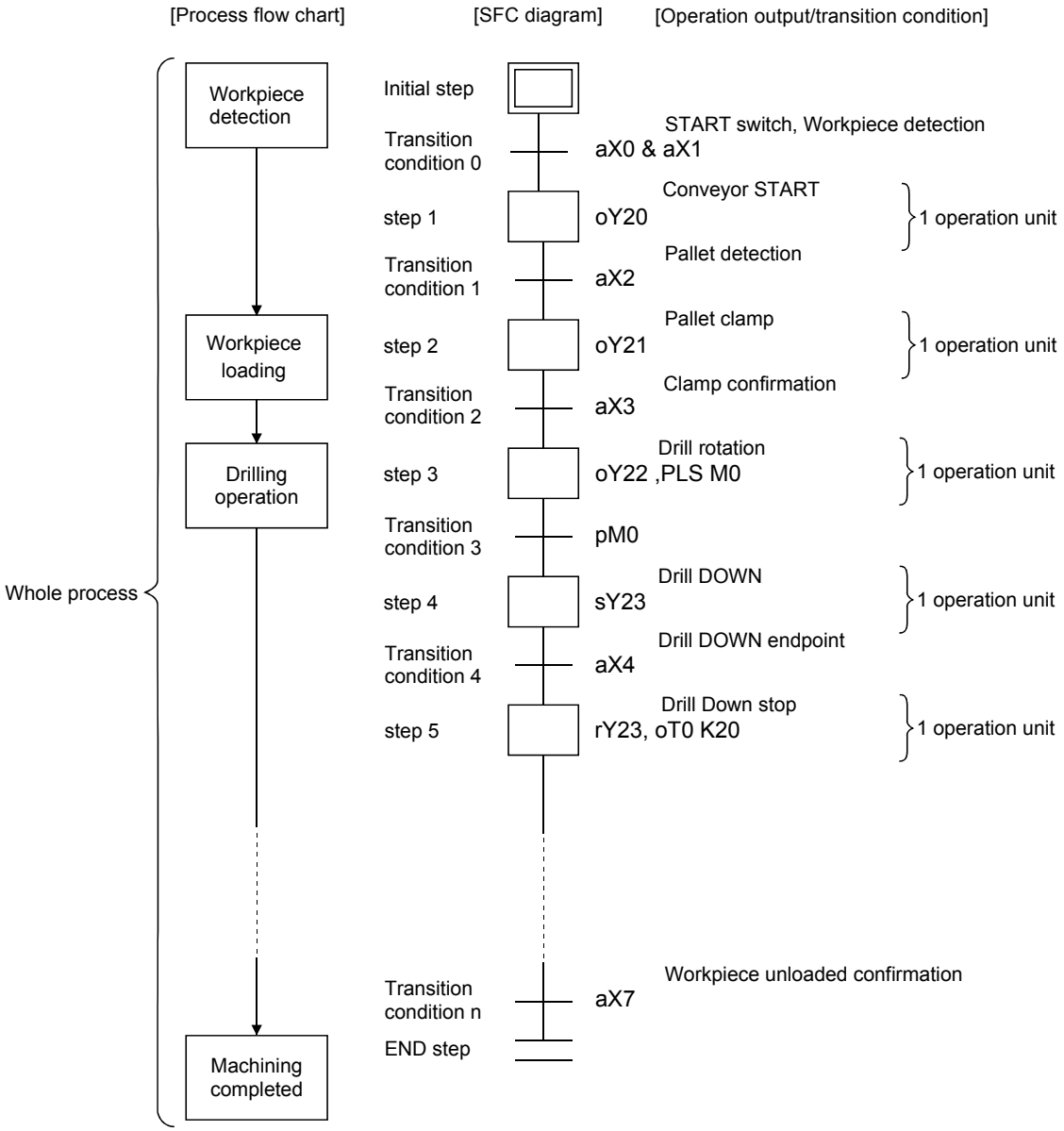


Commands equivalent to contacts cannot be described in the step.

1 - 2

1.1 SFC Program

The SFC program consists of steps that represent units of operations in a series of machine operations.
In each step, the actual detailed control is programmed by using a ladder circuit.
Grouping steps into one block in process units allows to create an SFC program that is capable of tracking all the processes as well as structuring the operation flow in each process.



The SFC program performs a sequence of operations, beginning from the “initial” step, proceeding to each subsequent step as the transition conditions are satisfied, and ending at the “END” step.

(1) When the SFC program is started, the “initial” step is executed first.

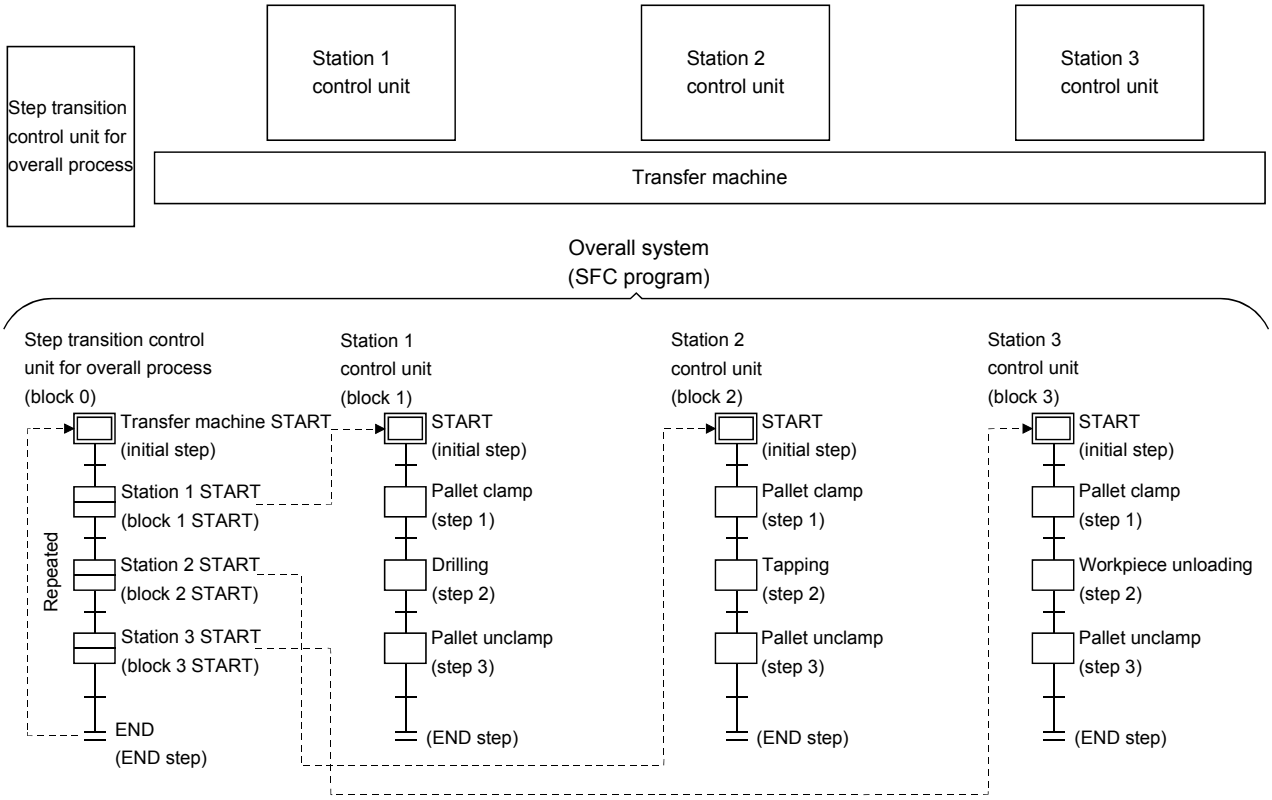
(2) Execution of the initial step continues until transition condition 0 is satisfied. When this transition condition is satisfied, execution of the initial step is stopped, and processing proceeds to the step which follows the initial step.

Processing of the SFC program continues from step to step in this manner until the END step has been executed.

1.2 SFC (MELSAP-L) Features

(1) Easy to design and maintain systems

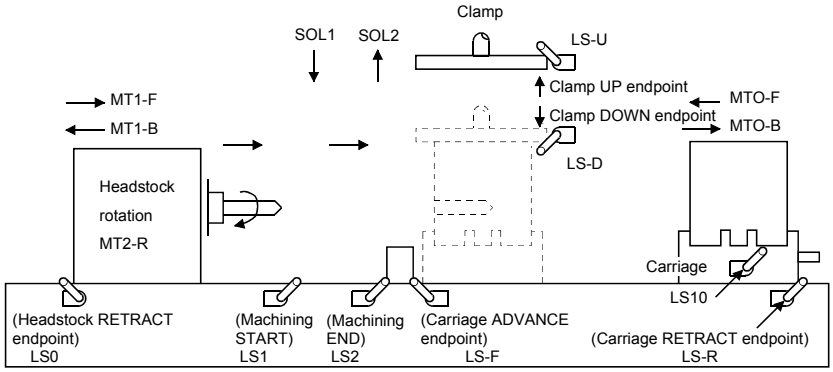
It is possible to correspond the controls of the entire facility, mechanical devices of each station, and all machines to the blocks and steps of the SFC program on a one-to-one basis. Because of this capability, systems can be designed and maintained with ease even by those with relatively little knowledge of sequence programs. Moreover, programs designed by other programmers using this format are much easier to decode than sequence programs.

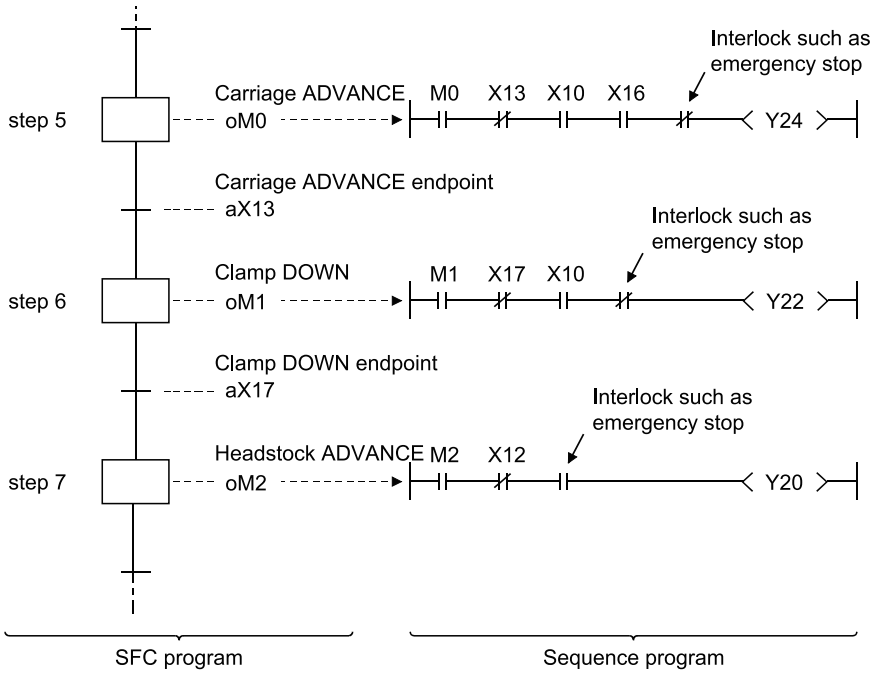


1 - 4

(2) Program development efficiency is enhanced by dividing control into parts

The machine control process can be divided into parts by describing the operation sequence and machine control separately. The MELSAP-L is used to describe the operation sequence for the machine, and a sequence program (circuit/list) is used to describe the machine control including individual interlock.

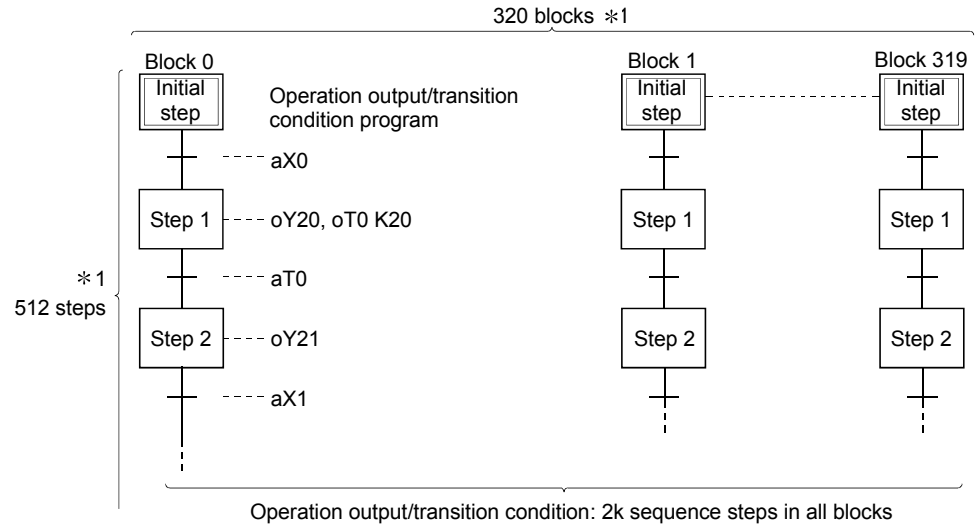




1 - 5

(3) Ease of division editing of blocks and steps according to control object

- A total of 320 blocks ＊1 can be created in a whole SFC program.
- Up to 512 steps ＊1 can be created in a single block.
- Up to 2k sequence steps of operation outputs/transition conditions can be created in all blocks. By dividing blocks and steps as shown below, tact time can be shortened and debugging/test operation can be performed easily.
- Blocks are divided properly according to the operation units of machines.
- Steps in each block are divided properly.



REMARKS

＊1: For the following CPU modules, 128 blocks and 128 steps can be created.

- Basic model QCPU
- Universal model QCPU (Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU)
- LCPU (L02SCPU, L02SCPU-P, L02CPU, L02CPU-P)

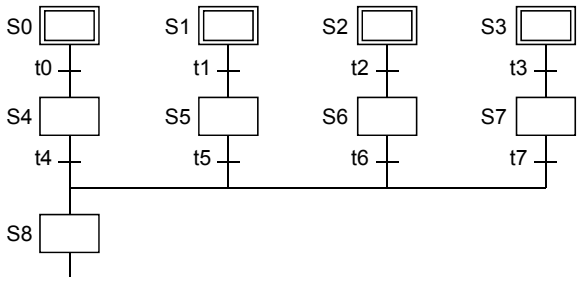
(4) Creation of multiple initial steps is possible

Multiple processes can easily be executed and combined. Initial steps are linked using a “selection coupling” format.

When multiple initial steps (S0 to S3) are active, the step where the transition condition (t4 to t7) immediately prior to the selected coupling is satisfied becomes inactive, and a transition to the next step occurs. Moreover, when the transition condition immediately prior to an active step is satisfied, the next step is executed in accordance with the parameter settings.

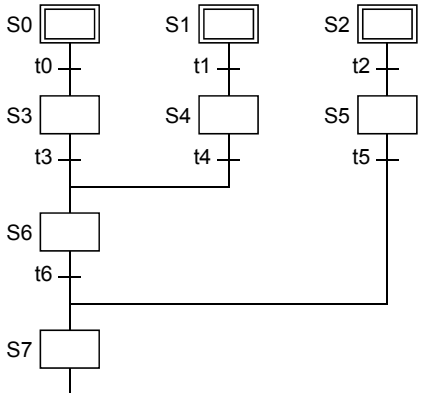
∗: The Basic model QCPU, Universal model QCPU, and LCPU cannot be selected in the parameter setting.
It operates in the default "Transfer" mode.

- Wait ..... Transition to the next step occurs after waiting for the next step to become inactive.
- Transfer..... Transition to the next step occurs even if the next step is active. (Default)
- Pause ..... An error occurs if the next step is active.



REMARKS

Linked steps can also be changed at each initial step.

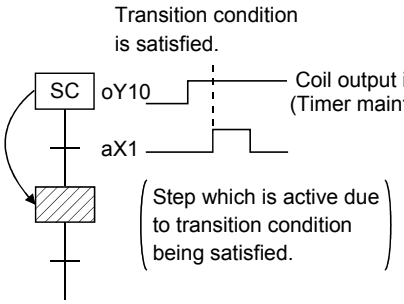


(5) Program design is easy due to a wealth of step attributes

A variety of step attributes can be assigned to each step. Used singly for a given control operation, or in combination, these attributes greatly simplify program design procedures.

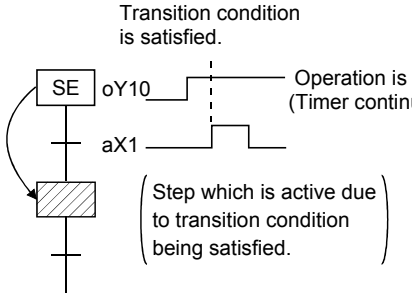
• Types of HOLD steps, and their operations

1) Coil HOLD step (SC)



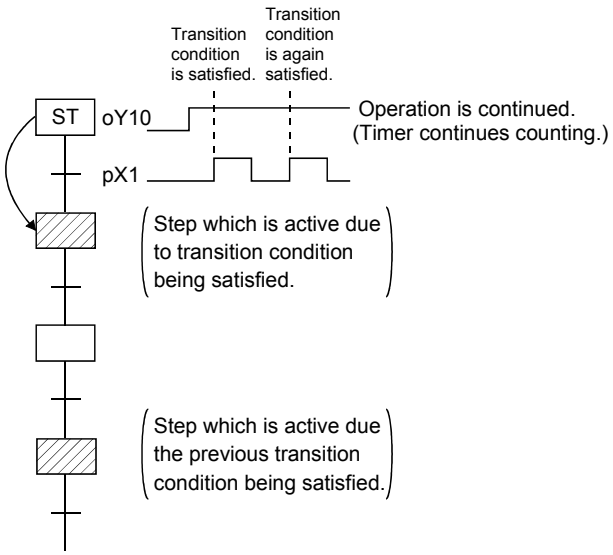
- After transition, the operation of the operation output is continued (put in HOLD status) and the coil output status when the transition condition is satisfied is maintained.
- Transition will not occur if the transition condition is satisfied again.
- Convenient for maintaining an output until the corresponding block is completed (hydraulic motor output, pass confirmation signal, etc.).
- When the output mode at block stop is OFF, it remains OFF after a block restart.

2) Operation HOLD step (no transition check) (SE)



- After transition, the operation of the operation output is continued (put in HOLD status).
- Transition will not occur if the transition condition is satisfied again.
- When the output mode at block stop is OFF, the operation is continued after a block restart, and therefore, the output is provided as a result of the operation that has been performed.

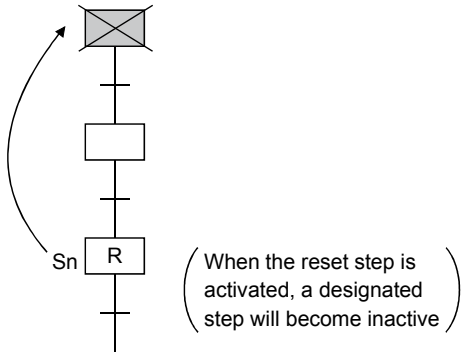
3) Operation HOLD step (with transition check) (ST)



- After transition, the operation of the operation output is continued (put in HOLD status).
- When the transition condition is satisfied again, transition is executed and the next step is reactivated.
- The operation of the operation output is executed at the reactivated next step, and when the transition condition is satisfied, transition occurs and the step is deactivated.

1 - 8

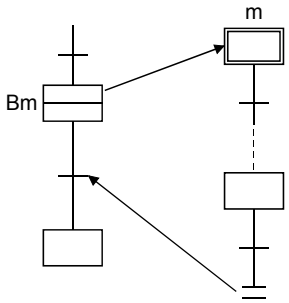
• Reset step (Sn R)



• When a HOLD status becomes unnecessary for machine control, or on selective branching to a manual ladder occurs after an error detection, etc., a reset request can be designated for the HOLD step, deactivating the step in question.

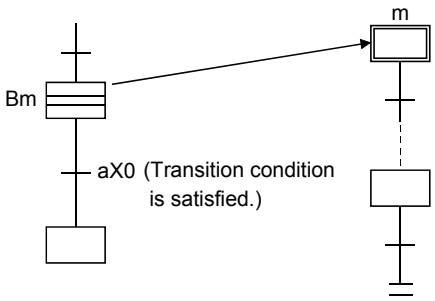
• Types of block START steps, and their operations

1) Block START step (with END check) (Bm ⊟)



• In the same manner as for a subroutine CALL-RET, a START source block transition will not occur until the end of the START destination block is reached.
• Convenient for starting the same block several times, or to use several blocks together, etc.
• A convenient way to return to the START source block and proceed to the next process block when a given process is completed in a processing line, for example.

2) Block START step (Without END check) (Bm ⊟)



• Even if the START destination block is active, a START source block transition occurs when the transition condition associated with the block START step is satisfied.
At this time, the processing of the START destination block will be continued unchanged until the end step is reached.
• By starting another block at a given step, the START destination block can be controlled independently and asynchronously with the START source block until processing of the current block is completed.

1 - 9

(6) A given function can be controlled in a variety of ways according to the application in question
Block functions such as START, END, temporary stop, restart, and forced activation and ending of specified steps can be controlled by SFC diagram symbols, SFC control instructions, or by SFC information registers.
• Control by SFC diagram symbols
..... Convenient for control of automatic operations with easy sequential control.
• Control by SFC instructions
..... Enables requests from program files other than the SFC, and is convenient for error processing, for example after emergency stops, and interrupt control.
• Control by SFC information devices
..... Enables control of SFC peripheral devices, and is convenient for partial operations such as debugging or trial runs.

Functions which can be controlled by these 3 methods are shown below.

Function	Control Method		
	SFC Diagram	SFC Control Instructions	SFC Information Devices
Block START (with END wait)	Bm ⊟	—	—
Block START (without END wait)	Bm ⊟	sBLm	Block START/END bit ON
Block END	⊥	rBLm	Block START/END bit OFF
Block STOP	—	PAUSE BLm	Block PAUSE/RESTART bit ON
Restart stopped block	—	RSTART BLm	Block PAUSE/RESTART bit OFF
Forced step activation	—	SSn SET BLm\Sn SCHG Kn	—
Forced step END	Sn R	RSn RST BLm\Sn SCHG Kn	—

1) In cases where the same function can be executed by a number of methods, the first control method which has been designated by the request output to the block or step in question will be the effective control method.
2) Functions controlled by a given control method can be canceled by another control method.
Example: For block START
The active block started by the SFC diagram (Bm ⊟) can be forcibly ended by executing the SFC control instruction (rBLm) before the END step (⊥) or by turning OFF the block START/END bit of the SFC information devices.

(7) The automatic scroll function helps the easy location of the spot where a mechanical problem occurs
The execution of an active (execution) block, an active (execution) step, an operation output/transition condition can be monitored (with the automatic scroll function) from peripheral devices.
Moreover, the transition watchdog function enables the detection of the step where transition does not occur after the designated time elapses.
These monitoring functions allow you to track down the spot where a problem occurs without a broad knowledge about sequence programs.

2. SYSTEM CONFIGURATION

(1) Applicable CPU modules

MELSAP-L (SFC programs) runs on the following CPU modules.

CPU module type	Model name	Restriction
Basic model QCPU	Q00JCPU, Q00CPU, Q01CPU	Modules whose serial number (first five digits) is 04122 or later
High Performance model QCPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	—
Process CPU	Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU	—
Redundant CPU	Q12PRHCPU, Q25PRHCPU	—
Universal model QCPU	Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDVCPU, Q03UDECPU, Q04UDHCPU, Q04UDVCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDVCPU, Q06UDEHCPU, Q10UCHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDVCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDVCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	—
LCPU	L02SCPU, L02SCPU-P, L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	—

2 - 1

(2) Peripheral devices for SFC programs

The following peripheral devices can be used to create, edit and monitor SFC programs.

Software package to be installed in a personal computer	CPU module					
	Basic model QCPU	High Performance model QCPU	Process CPU	Redundant CPU	Universal model QCPU	LCPU
SW4D5C-GPPW-E or later	×	○	×	×	×	×
GX Developer Version 7.10L (SW7D5C-GPPW-E) or later	×	○	△*2	×	×	×
GX Developer Version 8 (SW8D5C-GPPW-E) or later	○	○	△*2	×	×	×
GX Developer Version 8.18U (SW8D5C-GPPW-E) or later	○	○	△*2	○	×	×
GX Developer Version 8.48A (SW8D5C-GPPW-E) or later	○	○	△*2	○	△*1	×
GX Developer Version 8.62Q (SW8D5C-GPPW-E) or later	○	○	△*2	○	△*3	×
GX Developer Version 8.68W (SW8D5C-GPPW-E) or later	○	○	○	○	△*4	×
GX Developer Version 8.78G (SW8D5C-GPPW-E) or later	○	○	○	○	△*5	×
GX Developer Version 8.89T (SW8D5C-GPPW-E) or later	○	○	○	○	△*5	△*8
GX Works2 Version 1.24A (SW1DNC-GXW2-E) or later	×	○	×	×	△*5	△*8
GX Works2 Version 1.25B (SW1DNC-GXW2-E) or later	×	○	×	×	△*6	△*8
GX Works2 Version 1.56J (SW1DNC-GXW2-E) or later	×	○	×	×	△*6	△*7
GX Works2 Version 1.98C (SW1DNC-GXW2-E) or later	○	○	○	○	○	△*9
GX Works2 Version 1.492N (SW1DNC-GXW2-E) or later	○	○	○	○	○	○

○: Available, ×: Not available, △: Partly available

*1: Available only with the Q02UCPU, Q03UDCPU, Q04UDHCPU, and Q06UDHCPU

*2: Available only with the Q12PHCPU and Q25PHCPU

*3: Available only with the Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06CDHCPU, Q13UDCPU, and Q13UDHCPU

*4: Available only with the Q02UCPU, Q03UD(E)CPU, Q04UD(E)HCPU, Q06UD(E)HCPU, Q13UD(E)HCPU, and Q26UD(E)HCPU

*5: Available only with the Q00U(J)CPU, Q00UCPU, Q02UCPU, Q03UD(E)CPU, Q04UD(E)HCPU, Q06UD(E)HCPU, Q10UD(E)HCPU, Q13UD(E)HCPU, Q20UD(E)HCPU, and Q26UD(E)HCPU

*6: Available only with the Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDECPU, Q04UDHCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDEHCPU, Q50UDEHCPU, and Q100UDEHCPU

*7: Available only with the L02CPU, L02CPU-P, L26CPU-BT, and L26CPU-PBT

*8: Available only with the L02CPU and L26CPU-BT

*9: Available only with the L02SCPU, L02CPU, L02CPU-P, L06CPU, L26CPU, L26CPU-BT, and L26CPU-PBT

3. SPECIFICATIONS

This chapter explains the performance specifications of SFC programs.

3.1 Performance Specifications Related to SFC Programs

3.1.1 When the Basic model QCPU is used

(1) Table 3.1 indicates the performance specifications related to SFC programs.

Table 3.1 Performance Specifications Related to SFC Program

Item		Q00JCPU	Q00CPU	Q01CPU
SFC program	Capacity	Max. 8k steps	Max. 8k steps	Max. 14k steps
	Number of files	Scannable SFC program: 1 file ∗1		
	Number of blocks	Max. 128 blocks		
	Number of SFC steps	Max. 1024 steps for all blocks, max. 128 steps for one block		
	Number of branches	Max. 32		
	Number of concurrently active steps	Max. 1024 steps for all blocks Max. 128 steps for one block (including HOLD steps)		
	Number of operation output sequence steps	Max. 2k steps for all blocks ∗2 512 steps per step		
	Number of transition condition sequence steps	One ladder block only		

∗1: SFC program for program management (Section 5.2.3) cannot be created.

∗2: The maximum number of sequence steps per block depends on an instruction used for operation output or a note editing setting. The number of steps (2k steps) indicated in the table applies when "Unite (United Note)" is selected for note editing. Note that 2k sequence steps per block may not be secured when "Peripheral (Peripheral Note)" is selected. If note editing is not set, 2k sequence steps or more per block may be secured depending on an instruction used.

REMARKS

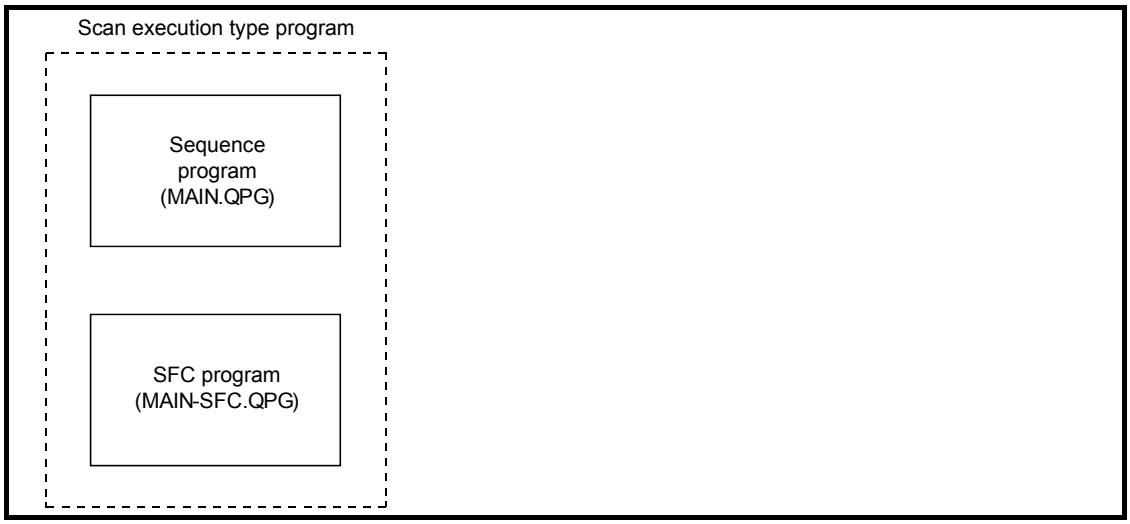
The step transition watchdog timer, STEP-RUN operation and step trace functions are not available.

3 - 1

(2) Precautions for creating SFC program

(a) Only one SFC program can be created.
The created SFC program is a "scan execution type program".

(b) The Basic model QCPU allows creation of a total of two program files: one SFC program and one sequence program.
(Two sequence programs or two SFC programs cannot be created.)



(c) The created sequence program and SFC program have the following file names. (The file names cannot be changed.)
- Sequence program: MAIN.QPG
- SFC program: MAIN-SFC.QPG

(d) The SFC program and sequence program are processed in order of "sequence program" and "SFC program".
(The processing order of the SFC program and sequence program cannot be changed.)

3 - 2

3.1.2 When the High Performance model QCPU, Process CPU, Redundant CPU, Universal model CPU, or LCPU is used

(1) Table 3.2 indicates the performance specifications related to SFC programs.

Table 3.2 Performance Specifications Related to SFC Programs

Item		Q02CPU Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
		Q02PHCPU	Q06PHCPU	Q12PHCPU	Q25PHCPU
		—	—	Q12PRHCPU	Q25PRHCPU
SFC program	Capacity	Max. 28k steps	Max. 60k steps	Max. 124k steps	Max. 252k steps
	Number of files	Scannable SFC program: 2 files (1 normal SFC program and 1 program execution management SFC program) ＊1			
	Number of blocks	Max. 320 blocks (0 to 319)			
	Number of SFC steps	Max. 8192 steps for all blocks, max. 512 steps for one block			
	Number of branches	Max. 32			
	Number of concurrently active steps (including HOLD steps)	Max. 1280 steps for all blocks Max. 256 steps for one block			
	Number of operation output sequence steps	Max. 2k steps for one block ＊2 512 steps for on one step			
	Number of transition condition sequence steps	One ladder block only			
Step transition watchdog timer function		Provided (10 timers)			

Table 3.2 Performance Specifications Related to SFC Programs

Item		Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU
SFC program	Capacity	Max. 10k steps		Max. 15k steps	Max. 20k steps
	Number of files	Scannable SFC program: 1 files (normal SFC program only)			
	Number of blocks	Max. 128 blocks (0 to 127)			
	Number of SFC steps	Max. 1024 steps for all blocks, max. 128 steps for one block			
	Number of branches	Max. 32			
	Number of concurrently active steps (including HOLD steps)	Max. 1024 steps for all blocks Max. 128 steps for one block			
	Number of operation output sequence steps	Max. 2k steps for all blocks ＊1 No restriction on one step			
	Number of transition condition sequence steps	One ladder block only			
Step transition watchdog timer function		None			

Table 3.2 Performance Specifications Related to SFC Programs

Item		Q03UD(E)CPU Q03UDVCPU	Q04UD(E)HCPU Q04UDVCPU	Q06UD(E)HCPU Q06UDVCPU	Q10UD(E)HCPU	Q13UD(E)HCPU Q13UDVCPU
SFC program	Capacity	Max. 30k steps	Max. 40k steps	Max. 60k steps	Max. 100k steps	Max. 130k steps
	Number of files	Scannable SFC program: 1 files (normal SFC program only)				
	Number of blocks	Max. 320 blocks (0 to 319)				
	Number of SFC steps	Max. 16384 steps for all blocks ＊3 ＊4 Max. 512 steps for one block				
	Number of branches	Max. 32				
	Number of concurrently active steps (including HOLD steps)	Max. 1280 steps for all blocks Max. 256 steps for one block				
	Number of operation output sequence steps	Max. 2k steps for all blocks ＊2 No restriction on one step				
	Number of transition condition sequence steps	One ladder block only				
Step transition watchdog timer function		None				

Table 3.2 Performance Specifications Related to SFC Programs

Item		Q20UD(E)HCPU	Q26UD(E)HCPU Q26UDVCPU	Q50UDEHCPU	Q100UDEHCPU
SFC program	Capacity	Max. 200k steps	Max. 260k steps	Max. 500k steps	Max. 1000k steps
	Number of files	Scannable SFC program: 1 files (normal SFC program only)			
	Number of blocks	Max. 320 blocks (0 to 319)			
	Number of SFC steps	Max. 16384 steps for all blocks ＊3 ＊4 Max. 512 steps for one block			
	Number of branches	Max. 32			
	Number of concurrently active steps (including HOLD steps)	Max. 1280 steps for all blocks Max. 256 steps for one block			
	Number of operation output sequence steps	Max. 2k steps for all blocks ＊2 No restriction on one step			
	Number of transition condition sequence steps	One ladder block only			
Step transition watchdog timer function		None			

＊1: Refer to Section 5.2.3 for the program execution management SFC program.

＊2: The maximum number of sequence steps per block depends on the instruction used for operation output or a note editing setting. The number of steps (2k steps) indicated in the table applies when "Unite (United Note)" is selected for note editing. Note that 2k sequence steps per block may not be secured when "Peripheral (Peripheral Note)" is selected.
If note editing is not set, 2k sequence steps or more per block may be secured depending on the instruction used.

∗3: For the Universal model QCPU whose serial number (first five digits) is "12051" or earlier, the maximum number of SFC steps is 8192 for all blocks.

∗4: For the Universal model QCPU whose serial number (first five digits) is "12052" or later, the maximum number of SFC steps can be changed by changing the step relay (S) points in the Device tab of the PLC parameter dialog box. For settings, refer to the QnUCPU User's Manual (Function Explanation, Program Fundamentals).

Table 3.2 Performance Specifications Related to SFC Programs

<table>
<tr><th colspan="2">Item</th><th>L02SCPU, L02SCPU-P,
L02CPU, L02CPU-P</th><th>L06CPU,
L06CPU-P</th><th>L26CPU,
L26CPU-P,
L26CPU-BT,
L26CPU-PBT</th></tr>
<tr><td rowspan="8">SFC program</td><td>Capacity</td><td>Max. 20k steps</td><td>Max. 60k steps</td><td>Max. 260k steps</td></tr>
<tr><td>Number of files</td><td colspan="3">Scannable SFC program: 1 file (normal SFC program only)</td></tr>
<tr><td>Number of blocks</td><td>Max. 128 blocks (0 to 127)</td><td colspan="2">Max. 320 blocks (0 to 319)</td></tr>
<tr><td>Number of SFC steps</td><td>Max. 1024 steps for all blocks, max. 128 steps for one block</td><td colspan="2">Max. 8192 steps for all blocks, max. 512 steps for one block</td></tr>
<tr><td>Number of branches</td><td colspan="3">Max. 32</td></tr>
<tr><td>Number of concurrently active steps (including HOLD steps)</td><td>Max. 1024 steps for all blocks, max. 128 steps for one block</td><td colspan="2">Max. 1280 steps for all blocks, max. 256 steps for one block</td></tr>
<tr><td>Number of operation output sequence steps</td><td colspan="3">Max. 2k steps for one block ∗1
No restriction on one step</td></tr>
<tr><td>Number of transition condition sequence steps</td><td colspan="3">One ladder block only</td></tr>
<tr><td colspan="2">Step transition watchdog timer function</td><td colspan="3">None</td></tr>
</table>

∗1: The maximum number of sequence steps per block depends on the instruction used for operation output or note editing setting. The number of steps (2k steps) indicated in the table applies when "Unite (United Note)" is selected for note editing. Note that 2k sequence steps per block may not be secured when "Peripheral (Peripheral Note)" is selected.
If note editing is not set, 2k sequence steps or more per block may be secured depending on the instruction used.

REMARKS

The STEP-RUN operation and step trace functions are not available.

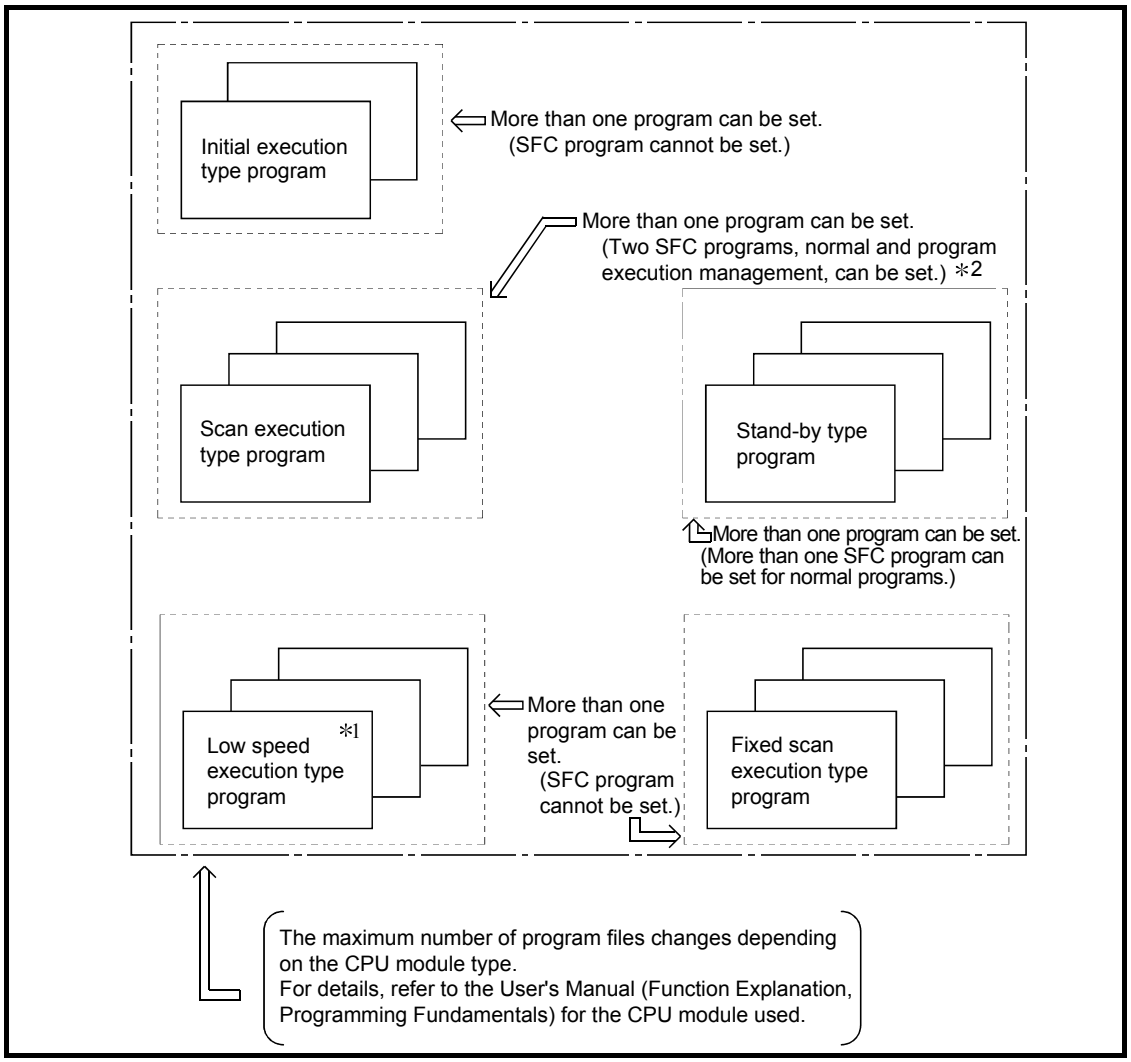
(2) Precautions for creating SFC program

(a) The SFC programs that can be created are "scan execution type program" and "stand-by type program".

(b) Two SFC programs (one normal SFC program and one program execution management SFC program) can be set as a scan execution type program. ∗2

(c) More than one SFC program can be set as a stand-by type program.

(d) The stand-by type SFC program is executed in the following procedure.
- The currently executed scan execution type program is switched to the stand-by type program.
- The stand-by type program to be executed is switched to the scan execution type program.



∗1: The Redundant CPU, Universal model QCPU, and LCPU do not support the low-speed execution type programs.

∗2: For the Universal model QCPU and LCPU, program execution management cannot be set.

REMARKS

Use the PSCAN or POFF instruction to switch the execution type of the program.
For details on the PSCAN and POFF instructions, refer to the Programming Manual (Common Instructions) for the CPU module used.

3.2 Device List

3.2.1 Device list of Basic model QCPU

Table 3.3 indicates the devices that can be used for the transition conditions and operation outputs of an SFC program.

Table 3.3 Device List

Classification	Type	Device name	Default			Parameter setting range
			Point	Range		
Internal user device	Bit device	Input	2048	X0 to X7FF	Hexadecimal	Can be changed within 16.4k words. *3
		Output	2048	Y0 to Y7FF	Hexadecimal	
		Internal relay	8192	M0 to M8191	Decimal	
		Latch relay	2048	L0 to L2047	Decimal	
		Annunciator	1024	F0 to F1023	Decimal	
		Edge relay	1024	V0 to V1023	Decimal	
		Step relay	2048	S0 to S127/block	Decimal	
		Link relay	2048	B0 to B7FF	Hexadecimal	
		Link special relay	1024	SB0 to SB3FF	Hexadecimal	
	Word device	Timer *1	512	T0 to T511	Decimal	
		Retentive timer *1	0	(ST0 to ST511)	Decimal	
		Counter *1	512	C0 to C511	Decimal	
		Data register	11136	D0 to D11135	Decimal	
		Link register	2048	W0 to W7FF	Hexadecimal	
		Link special register	1024	SW0 to SW3FF	Hexadecimal	
Internal system device	Bit device	Function input	16	FX0 to FXF	Hexadecimal	N/A
		Function output	16	FY0 to FYF	Hexadecimal	
		Special relay	1024	SM0 to SM1023	Decimal	
	Word device	Function register	5	FD0 to FD4	Decimal	
		Special register	1024	SD0 to SD1023	Decimal	
Link direct device	Bit device	Link input	8192	Jn\X0 to Jn\X1FFF	Hexadecimal	N/A
		Link output	8192	Jn\Y0 to Jn\Y1FFF	Hexadecimal	
		Link relay	16384	Jn\B0 to Jn\B3FFF	Hexadecimal	
		Link special relay	512	Jn\SB0 to Jn\SB1FF	Hexadecimal	
	Word device	Link register	16384	Jn\W0 to Jn\W3FFF	Hexadecimal	
		Link special register	512	Jn\SW0 to Jn\SW1FF	Hexadecimal	
Module access device	Word device	Intelligent function module device	65536	Un\G0 to Un\G65535 *2	Decimal	N/A
Index register	Word device	Index register	10	Z0 to Z9	Decimal	N/A

(Continued to the next page)

Table 3.3 Device List (continued)

Classification	Type	Device name		Default			Parameter setting range
				Point	Range		
File register *5	Word device	File register		64k	• R0 to R32767 • ZR0 to ZR65535	Decimal	N/A
Nesting	—	Nesting		15	N0 to N14	Decimal	N/A
Pointer	—	Pointer		300	P0 to P299	Decimal	N/A
		Interrupt pointer		128	I0 to I127	Decimal	
Others	Bit device	SFC block device		128	BL0 to BL127	Decimal	N/A
	—	Network No. specification device		239	J1 to J239	Decimal	N/A
		I/O No. specification device	Q00JCPU	—	U0 to UF	Hexadecimal	
			Q00CPU, Q01CPU	—	U0 to U3F	Hexadecimal	
	—	Macro instruction argument device		—	VD0 to VD□	Decimal	N/A
Constant	—	Decimal constant		K-2147483648 to 2147483647			
		Hexadecimal constant		H0 to HFFFFFFFF			
		Real constant		E±1.17550-38 to E±3.40282+38			
		Character string constant		"ABC", "123" *4			

*1: For the timer, retentive timer, and counter, contact/coil values are stored in bit devices, and current values are stored in word devices.

*2: The number of points that can be actually used varies depending on the intelligent function module. For the points in the buffer memory, refer to the manual for the intelligent function module used.

*3: The value can be changed in the PLC parameter dialog box of GX Developer.
(Except for input, output, step relay, link special relay, and link special register. Refer to Section 9.2.)

*4: Character strings can be used only for the $MOV, STR, DSTR, VAL, DVAL, ESTR, and EVAL instructions.
They cannot be used for the other instructions.

*5: Because the Q00JCPU does not have the standard RAM, the file register cannot be used.

3 - 9

3.2.2 Device list of High Performance model QCPU, Process CPU, and Redundant CPU

Table 3.4 indicates the devices that can be used for the transition conditions and operation outputs of SFC programs.

Table 3.4 Device List

Classification	Type	Device name	Default			Parameter setting range
			Point	Range		
Internal user device	Bit device	Input	8192	X0 to X1FFF	Hexadecimal	Can be changed within 29k words. *3
		Output	8192	Y0 to Y1FFF	Hexadecimal	
		Internal relay	8192	M0 to M8191	Decimal	
		Latch relay	8192	L0 to L8191	Decimal	
		Annunciator	2048	F0 to F2047	Decimal	
		Edge relay	2048	V0 to V2047	Decimal	
		Step relay	8192	S0 to S511/block	Decimal	
		Link relay	8192	B0 to B1FFF	Hexadecimal	
		Link special relay	2048	SB0 to SB7FF	Hexadecimal	
	Word device	Timer *1	2048	T0 to T2047	Decimal	
		Retentive timer *1	0	(ST0 to ST2047)	Decimal	
		Counter *1	1024	C0 to C1023	Decimal	
		Data register	12288	D0 to D12287	Decimal	
		Link register	8192	W0 to W1FFF	Hexadecimal	
		Link special register	2048	SW0 to SW7FF	Hexadecimal	
Internal system device	Bit device	Function input	16	FX0 to FXF	Hexadecimal	N/A
		Function output	16	FY0 to FYF	Hexadecimal	
		Special relay	2048	SM0 to SM2047	Decimal	
	Word device	Function register	5	FD0 to FD4	Decimal	
		Special register	2048	SD0 to SD2047	Decimal	
Link direct device	Bit device	Link input	8192	Jn\X0 to Jn\X1FFF	Hexadecimal	N/A
		Link output	8192	Jn\Y0 to Jn\Y1FFF	Hexadecimal	
		Link relay	16384	Jn\B0 to Jn\B3FFF	Hexadecimal	
		Link special relay	512	Jn\SB0 to Jn\SB1FF	Hexadecimal	
	Word device	Link register	16384	Jn\W0 to Jn\W3FFF	Hexadecimal	
		Link special register	512	Jn\SW0 to Jn\SW1FF	Hexadecimal	
Module access device	Word device	Intelligent function module device	65536	Un\G0 to Un\G65535 *2	Decimal	N/A
		Cyclic transmission area device *4	14336	U3En\G0 to U3En\G4095	Decimal	Setting available
Index register	Word device	Index register	20	Z0 to Z15	Decimal	N/A

(Continued to the next page)

Table 3.4 Device List (continued)

Classification	Type	Device name	Default			Parameter setting range
			Point	Range		
File register	Word device	File register	0	—	—	0 to 1018k points
Nesting	—	Nesting	15	N0 to N14	Decimal	N/A
Pointer	—	Pointer	4096	P0 to P4095	Decimal	N/A
		Interrupt pointer	256	I0 to I255	Decimal	
Others	Bit device	SFC block device	320	BL0 to BL319	Decimal	N/A
	—	Network No. specification device	512	TR0 to TR511	Decimal	
		I/O No. specification device	255	J1 to J255	Hexadecimal	
		Macro instruction argument device	—	U0 to UFF	Hexadecimal	
Constant	—	Decimal constant	K-2147483648 to 2147483647			
		Hexadecimal constant	H0 to HFFFFFFFF			
		Real constant	Single-precision floating-point data: E±1.17549435-38 to E±3.40282347+38			
			Double-precision floating-point data: E±2.2250738585072014-308 to E±1.7976931348623157+308			
		Character string constant	"ABC", "123"			

*1: For the timer, retentive timer, and counter, contact/coil values are stored in bit devices, and current values are stored in word devices.

*2: The number of points that can be actually used varies depending on the intelligent function module. For the points in the buffer memory, refer to the manual for the intelligent function module or special function module used.

*3: The value can be changed in the Device setting in the PLC parameter dialog box. (Except for input, output, step relay, link special relay, and link special register. Refer to Section 9.2.)

3.2.3 Device list of Universal model QCPU

Table 3.5 indicates the devices that can be used for the transition conditions and operation outputs of SFC programs.

Table 3.5 Device List

Classification	Type	Device name	Default			Parameter setting range
			Point	Range		
Internal user device	Bit device	Input	8192	X0 to X1FFF	Hexadecimal	Can be changed within 29k words. *3 *19
		Output	8192	Y0 to Y1FFF	Hexadecimal	
		Internal relay	8192 *20	M0 to M8191 *21	Decimal	
		Latch relay	8192	L0 to L8191	Decimal	
		Annunciator	2048	F0 to F2047	Decimal	
		Edge relay	2048	V0 to V2047	Decimal	
		Step relay	8192	S0 to S8191	Decimal	
		Link relay	8192	B0 to B1FFF	Hexadecimal	
		Link special relay	2048	SB0 to SB7FF	Hexadecimal	
	Word device	Timer *1	2048	T0 to T2047	Decimal	
		Retentive timer *1	0	(ST0 to ST2047)	Decimal	
		Counter *1	1024	C0 to C1023	Decimal	
		Data register	12288 *22	D0 to D12287 *23	Decimal	
		Link register	8192	W0 to W1FFF	Hexadecimal	
		Link special register	2048	SW0 to SW7FF	Hexadecimal	
Internal system device	Bit device	Function input	16	FX0 to FXF	Hexadecimal	N/A
		Function output	16	FY0 to FYF	Hexadecimal	
		Special relay	2048	SM0 to SM2047	Decimal	
	Word device	Function register	5	FD0 to FD4	Decimal	
		Special register	2048	SD0 to SD2047	Decimal	
Link direct device	Bit device	Link input	16384 *14	Jn\X0 to Jn\X3FFF *15	Hexadecimal	N/A
		Link output	16384 *14	Jn\Y0 to Jn\Y3FFF *15	Hexadecimal	
		Link relay	32768	Jn\B0 to Jn\B7FFF	Hexadecimal	
		Link special relay	512	Jn\SB0 to Jn\SB1FF	Hexadecimal	
	Word device	Link register	131072	Jn\W0 to Jn\W1FFFF	Hexadecimal	
		Link special register	512	Jn\SW0 to Jn\SW1FF	Hexadecimal	

(Continued to the next page)

3 - 12

Table 3.5 Device List (continued)

Classification	Type	Device name	Default			Parameter setting range
			Point	Range		
Module access device	Word device	Intelligent function module device	65536	Un\G0 to Un\G65535 *2	Decimal	N/A
		Cyclic transmission area device *4	4096	U3En\G0 to U3En\G4095	Decimal	N/A
			14336	U3En\G10000 to U3En\G24335	Decimal	Setting available
Index register/standard device register	Word device	Index register/standard device register	20	Z0 to Z19	Decimal	N/A
File register *7	Word device	File register	0	—	—	0 to 4086k points *6 *24
Extended data register *7	Word device	Extended data register	0 *16	—	—	
Extended link register *7	Word device	Extended link register	0	—	—	
Nesting	—	Nesting	15	N0 to N14	Decimal	N/A
Pointer	—	Pointer	4096 *8 *17	P0 to P4095 *9 *18	Decimal	N/A
		Interrupt pointer	256 *10	I0 to I255 *11	Decimal	
Others	Bit device	SFC block device	320 *25	BL0 to BL319 *12	Decimal	N/A
	—	Network No. Specification device	255	J1 to J255	Decimal	
		I/O No. specification device	516	U0 to UFF, U3E0 to U3E3 *13	Hexadecimal	
		Macro instruction argument device	10	VD0 to VD9	Decimal	
Constant	—	Decimal constant	K-2147483648 to 2147483647			
		Hexadecimal constant	H0 to HFFFFFFFF			
		Real constant	Single-precision floating-point data: E±1.17549435-38 to E±3.40282347+38			
			Double-precision floating-point data *5: E±2.2250738585072014-308 to E±1.7976931348623157+308			
		Character string constant	"ABC", "123"			

*1: For the timer, retentive timer, and counter, contact/coil values are stored in bit devices, and current values are stored in word devices.

*2: The number of points that can be actually used varies depending on the intelligent function module. For the points in the buffer memory, refer to the manual for the intelligent function module used.

*3: The number of points can be changed (except for input, output, and step relay) in the Device tab of the PLC parameter dialog box.
Note that the step relay points can be changed to 0 point for the Universal model QCPU whose serial number (first five digits) is "10042" or later.
For the Universal model QCPU whose serial number (first five digits) is "12052" or later, the step relay points can be set in increments of 1k points and up to the following points.
- Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 8192 points
- Universal model QCPUs other than the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU: 16384 points

*4: Available only in a multiple CPU system configuration.
*5: Up to 15 digits can be entered in GX Developer.
*6: The total of the points for the file register, extended data register (D), and extended link register (W)
*7: The device cannot be used on the Q00UJCPU.
*8: For the Q00UJCPU, Q00UCPU, and Q01UCPU, the number of points is 512.
*9: For the Q00UJCPU, Q00UCPU, and Q01UCPU, the range is P0 to P511.
*10: For the Q00UJCPU, Q00UCPU, and Q01UCPU, the number of points is 128.
*11: For the Q00UJCPU, Q00UCPU, and Q01UCPU, the range is I0 to I127.
*12: For the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU, the range is BL0 to BL127.
*13: The range differs depending on the CPU module: U0 to UF for the Q00UJCPU; U0 to U3F and U3E0 to 3E2 for the Q00UCPU and Q01UCPU; and U0 to U7F and U3E0 to U3E2 for the Q02UCPU.
*14: For the Universal model QCPU whose serial number (first five digits) is "12011" or earlier, the number of points is 8192.
*15: For the Universal model QCPU whose serial number (first five digits) is "12011" or earlier, the range is Jn\X/Y0 to Jn\1FFF.
*16: For the Q50UDEHCPU and Q100UDEHCPU, the number of points is 128k.
*17: For the Q50UDEHCPU and Q100UDEHCPU, the points is 8192.
*18: For the Q50UDEHCPU and Q100UDEHCPU, the range is P0 to P8191.
*19: The changeable range differs depending on the CPU module: within 30k words for the Q03UDVCPU; within 40k words for the Q04UDVCPU and Q06UDVCPU; and within 60k words for the Q13UDVCPU and Q26UDVCPU.
*20: The number of points differs depending on the CPU module: 9216 for the Q03UDVCPU; 15360 for the Q04UDVCPU and Q06UDVCPU; and 28672 for the Q13UDVCPU and Q26UDVCPU.
*21: The range differs depending on the CPU module: M0 to M9215 for the Q03UDVCPU; M0 to M15359 for the Q04UDVCPU and Q06UDVCPU; and M0 to M28671 for the Q13UDVCPU and Q26UDVCPU.
*22: The number of points differs depending on the CPU module: 13312 for the Q03UDVCPU; 22528 for the Q04UDVCPU and Q06UDVCPU; and 41984 for the Q13UDVCPU and Q26UDVCPU.
*23: The range differs depending on the CPU module: D0 to D13311 for the Q03UDVCPU; D0 to D22527 for the Q04UDVCPU and Q06UDVCPU; and D0 to D41983 for the Q13UDVCPU and Q26UDVCPU.
*24: The setting range differs depending on the CPU module: 0 to 4192k points for the Q03UDVCPU, 0 to 4224k points for the Q04UDVCPU, 0 to 4480k points for Q06UDVCPU, 0 to 4608k points for the Q13UDVCPU, and 0 to 4736k points for the Q26UDVCPU.
*25: For the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU, the number of points is 128.

3.2.4 Device list of LCPU

Table 3.6 indicates the devices that can be used for the transition conditions and operation outputs of SFC programs.

Table 3.6 Device List

Classification	Type	Device name	Default			Parameter setting range
			Point	Range		
Internal user device	Bit device	Input	8192	X0 to X1FFF	Hexadecimal	N/A
		Output	8192	Y0 to Y1FFF	Hexadecimal	
		Internal relay	8192	M0 to M8191	Decimal	Setting available(Up to 29K words for the internal user device)*6
		Latch relay	8192	L0 to L8191	Decimal	
		Link relay	8192	B0 to B1FFF	Hexadecimal	
		Annunciator	2048	F0 to F2047	Decimal	
		Link special relay	2048	SB0 to SB7FF	Hexadecimal	
		Edge relay	2048	V0 to V2047	Decimal	
		Step relay	8192	S0 to S8191	Decimal	
	•Bit device (contact/coil) •Word device (current value)	Timer	2048	T0 to T2047	Decimal	
		Retentive counter	0	(ST0 to ST2047)	Decimal	
		Counter	1024	C0 to C1023	Decimal	
	Word device	Data register	12288	D0 to D12287	Decimal	
		Link register	8192	W0 to W1FFF	Hexadecimal	
		Link special register	2048	SW0 to SW7FF	Hexadecimal	
Internal system device	Bit device	Function input	16	FX0 to FXF	Hexadecimal	N/A
		Function output	16	FY0 to FYF	Hexadecimal	
		Special relay	2048	SM0 to SM2047	Decimal	
	Word device	Function register	5	FD0 to FD4	Decimal	
		Special register	2048	SD0 to SD2047	Decimal	
Module access device	Word device	Intelligent function module device	65536	Un\G0 to Un\G65535 *2	Decimal	N/A
Index register/standard device register	Word device	Index register/standard device register	20	Z0 to Z19	Decimal	N/A

(Continued to the next page)

3 - 15

Table 3.6 Device List (continued)

Classification	Type	Device name	Default			Parameter setting range
			Point	Range		
File register	Word device	File register	0	—	Decimal	0 to 384K points in total *3 (in 1K units)
Extended data register	Word device	Extended data register	128K	D12288 to D143359*1	Decimal	
Extended link register	Word device	Extended link register	0	—	Hexadecimal	
Nesting	—	Nesting	15	N0 to N14	Decimal	N/A
Pointer	—	Pointer	4096*7	P0 to P4095*7	Decimal	N/A*7
		Interrupt pointer	256	I0 to I255	Decimal	N/A
Others	Bit device	SFC block device	320	BL0 to BL319 points *4	Decimal	N/A
	—	I/O No. specification device	—	U0 to UFF*5	Decimal	
		Macro instruction argument device	10	VD0 to VD9	Decimal	

*1: For the L02SCPU, L02SCPU-P, L02CPU, L02CPU-P, the number of points is 32K (D12288 to D45055).

*2: The number of points that can be actually used varies depending on the intelligent function module. Refer to the manual for each intelligent function module.

*3: For the L02SCPU, L02SCPU-P, L02CPU, and L02CPU-P, the total number of points is 0 to 64K.

*4: For the L02SCPU, L02SCPU-P, L02CPU, and L02CPU-P, the number of points is 128 (BL0 to B127).

*5: For the L02SCPU, L02SCPU-P, L02CPU, and L02CPU-P, the range is U0 to U3F.

*6: For the LCPU whose serial number (first five digits) is "15101" or earlier, either 0K point or 8K point can be set for the step relay. For the LCPU whose serial number (first five digits) is "15102" or later, the step relay points can be set up to the following points.

• L02(S)CPU, L02(S)CPU-P: 8192 points

• Other models: 16384 points

*7: For the L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, and L26CPU-PBT whose serial number (first five digits) is "16042" or later, the pointer for automatic-assign device is extended up to 32768 points in the Device tab of the PLC parameter dialog box. For details, refer to the MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals).

3 - 16

3.3 Processing Time

3.3.1 Processing time for SFC program

The time required to process the SFC program is discussed below.

(1) Method for calculating the SFC program processing time
Calculate the SFC program processing time with the following expression

SFC program processing time = (A) + (B) + (C)

(a) "(A): Processing time of operation outputs in all blocks"
Indicates the total sum of the processing times of the instructions used for the operation outputs of all steps that are active.
For the processing times of the instructions, refer to the QCPU Programming Manual (Common Instructions).

(b) "(B): Processing time of all transition conditions"
Indicates the total sum of the processing times of the instructions used for the transition conditions associated with all steps that are active.
For the processing times of the instructions, refer to the QCPU Programming Manual (Common Instructions).

(c) "(C)" SFC system processing time"
Calculate the SFC system processing time with the following expression.

SFC system processing time = (a) + (b) + (c) + (d) + (e) + (f) + (g)

Processing Time		Calculation of Processing Time (Unit: μs)
(a)	Active block processing time	(Active block processing time) = (active block processing time coefficient) × (number of active blocks) • Active block processing time: System processing time required to execute active blocks • Number of active blocks: Number of blocks that are active
(b)	Inactive block processing time	(Inactive block processing time) = (inactive block processing time coefficient) × (number of inactive blocks) • Inactive block processing time: System processing time required to execute inactive blocks • Number of inactive blocks: Number of blocks that are inactive
(c)	Nonexistent block processing time	(Nonexistent block processing time) = (nonexistent block processing time coefficient) × (number of nonexistent blocks) • Nonexistent block processing time: System processing time required to execute blocks that have not been created • Number of nonexistent blocks: Number of blocks where programs have not been created within the number of blocks set in the parameter
(d)	Active step processing time	(Active step processing time) = (active step processing time coefficient) × (number of active steps) • Active step processing time: Time required to execute active steps • Number of active steps: Number of steps that are active in all blocks
(e)	Active transition processing time	(Active transition processing time) = (active transition processing time coefficient) × (number of active transitions) • Active transition processing time: System processing time required to execute active transitions • Number of active transitions: Number of transition conditions associated with all steps that are active in all blocks
(f)	Transition condition-satisfied step processing time	(Transition condition-satisfied step processing time) = (transition condition-satisfied step processing time coefficient) × (number of transition condition-satisfied steps) • Transition condition-satisfied step processing time: Time required to perform OFF execution of active steps • Number of transition condition-satisfied steps: Number of steps where operation outputs are turned OFF since transition conditions were satisfied in all blocks
(g)	SFC end processing time	(SFC end processing time) = (SFC end processing time) • SFC end processing time: System processing time required to perform the end processing of SFC program.

(2) System processing times for different CPU module models

(a) When Basic model QCPU is used

Item		Q00JCPU	Q00CPU	Q01CPU
Active block processing time coefficient		41.9μs	35.5μs	27.3μs
Inactive block processing time coefficient		10.5μs	8.8μs	6.8μs
Nonexistent block processing time coefficient		1.1μs	0.9μs	0.7μs
Active step processing time coefficient		31.6μs	26.7μs	20.5μs
Active transition processing time coefficient		10.2μs	8.7μs	6.7μs
Transition condition-satisfied step processing time coefficient	With HOLD step designation＊	216.0μs	182.8μs	140.6μs
	Normal step designation	263.5μs	222.9μs	171.5μs
SFC end processing time		66.8μs	56.5μs	43.5μs

(b) When High Performance model QCPU, Process CPU or Redundant CPU is used

Item		High Performance model QCPU		Process CPU	Redundant CPU
		QnCPU	QnHCPU	QnPHCPU	QnPRHCPU
Active block processing time coefficient		33.7μs	14.5μs	14.5μs	14.5μs
Inactive block processing time coefficient		12.0μs	5.2μs	5.2μs	5.2μs
Nonexistent block processing time coefficient		4.1μs	1.8μs	1.8μs	1.8μs
Active step processing time coefficient		24.5μs	10.6μs	10.6μs	10.6μs
Active transition processing time coefficient		10.0μs	4.3μs	4.3μs	4.3μs
Transition condition-satisfied step processing time coefficient	With HOLD step designation ＊	130.4μs	56.2μs	56.2μs	56.2μs
	Normal step designation	119.4μs	51.5μs	51.5μs	51.5μs
SFC end processing time		108.2μs	46.6μs	46.6μs	46.6μs

(c) When Universal model QCPU is used

Item		Universal model QCPU			
		Q00UJCPU, Q00UCPU, Q01UCPU	Q02UCPU	Q03UDCPU, Q03UDECPU	Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU
Active block processing time coefficient		12.7μs	8.4μs	8.3μs	7.0μs
Inactive block processing time coefficient		5.3μs	3.9μs	3.8μs	3.4μs
Nonexistent block processing time coefficient		0.9μs	0.8μs	0.7μs	0.6μs
Active step processing time coefficient		11.9μs	8.6μs	8.2μs	6.4μs
Active transition processing time coefficient		3.4μs	2.1μs	2.0μs	1.6μs
Transition condition-satisfied step processing time coefficient	With HOLD step designation＊	86.7μs	69.6μs	60.3μs	42.7μs
	Normal step designation	106.9μs	83.2μs	73.7μs	52.0μs
SFC end processing time		67.5μs	38.4μs	36.6μs	26.9μs

＊: The HOLD step includes all of the coil hold steps and operation hold steps (with or without transition check).
The Normal step represents steps other than the above.

Item		Universal model QCPU	
		Q03UDVCPU	Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, Q26UDVCPU
Active block processing time coefficient		5.0μs	4.8μs
Inactive block processing time coefficient		2.5μs	2.3μs
Nonexistent block processing time coefficient		0.60μs	0.33μs
Active step processing time coefficient		5.8μs	5.5μs
Active transition processing time coefficient		1.3μs	1.3μs
Transition condition-satisfied step processing time coefficient	With HOLD step designation ＊	38μs	34μs
	Normal step designation	50μs	41μs
SFC end processing time		25.5μs	25.5μs

(d) When LCPU is used

Item		L02SCPU, L02SCPU-P	L02CPU, L02CPU-P	L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT
Active block processing time coefficient		12.7μs	8.5μs	7.0μs
Inactive block processing time coefficient		5.3μs	3.8μs	3.4μs
Nonexistent block processing time coefficient		0.9μs	1.2μs	0.6μs
Active step processing time coefficient		11.9μs	8.7μs	6.4μs
Active transition processing time coefficient		3.4μs	2.0μs	1.6μs
Transition condition-satisfied step processing time coefficient	With HOLD step designation＊	86.7μs	66.1μs	42.7μs
	Normal step designation	106.9μs	79.4μs	52.0μs
SFC end processing time		67.5μs	44.7μs	26.9μs

＊: The HOLD step includes all of the coil hold steps and operation hold steps (with or without transition check).
The Normal step represents steps other than the above.

3 - 19

[SFC system processing time calculation example]

Using the Q25HCPU as an example, the processing time for the SFC system is calculated as shown below, given the following conditions.

- Designated at initial START
- Number of active blocks: 30 (active blocks at SFC program)
- Number of inactive blocks: 70 (inactive blocks at SFC program)
- Number of nonexistent blocks: 50 (number of blocks between 0 and the max. created block No. which have no SFC program)
- Number of active steps: 60 (active steps within active blocks)
- Active step transition conditions: 60
- Steps with satisfied transition conditions: 10
  (active steps (no HOLD steps) with satisfied transition conditions)

SFC system process time =(14.5 × 30) + (5.2 × 70) + (1.8 × 50)
+ (10.6 × 60) + (4.3 × 60) + (56.2 × 10) + 46.6
= 2391.6 μs ≒ 2.40 ms

In this case, calculation using the equation shown above results in an SFC system processing time of 2.40 ms.

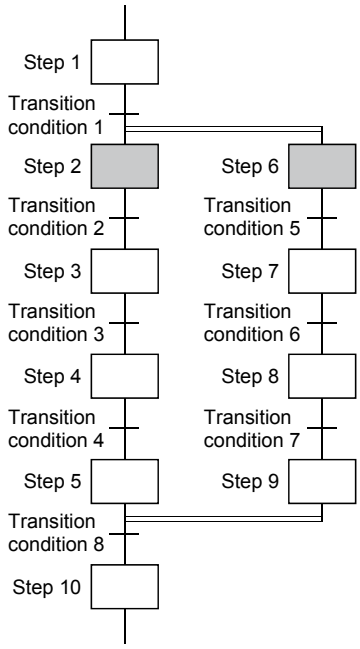
The scan time is the total of the following times;
SFC system processing time, main sequence program processing time, SFC active step transition condition time, and CPU END processing time.

The scan time is the total of the following times:
SFC system processing time, main sequence program processing time, processing time of ladder circuit having transition conditions associated with SFC's active steps, and CPU module's END processing time.

The number of active steps, the number of transition conditions, and the number of steps with satisfied transition conditions varies according to the conditions shown below.

- When transition condition is unsatisfied
- When transition condition is satisfied (without continuous transition)
- When transition condition is satisfied (with continuous transition)

The method for determining the number of the above items is illustrated in the SFC diagram below.



3 - 20

The following table indicates the number of active steps, number of active transitions, and number of transition condition-satisfied steps when Step 2 and Step 6 are active.

Whether Transition Conditions Are Satisfied or Not	Presence/Absence of Continuous Transition	Number of Active Steps	Number of Active Transitions	Number of Transition Condition-Satisfied Steps
• Transition conditions not satisfied	—	2 (Steps 2, 6)	2 (Transition conditions 2, 5)	0
• Transition conditions 2, 5 satisfied • Transition conditions 3, 6 not satisfied	Absence	2 (Steps 2, 6)	2 (Transition conditions 2, 5)	2 (Steps 2, 6)
	Presence	4 (Steps 2, 3, 6, 7)	4 (Transition conditions 2, 3, 5, 6)	2 (Steps 2, 6)
• Transition conditions 2, 3, 5, 6 satisfied	Absence	2 (Steps 2, 6)	2 (Transition conditions 2, 5)	2 (Steps 2, 6)
	Presence	6 (Steps 2 to 4, 6 to 8)	6 (Transition conditions 2 to 7)	4 (Steps 2, 3, 6, 7)

3 - 21

3.3.2 Processing time for S(P).SFCSCOMR instruction and S(P).SFCTCOMR instruction

Processing time for S(P).SFCSCOMR instruction and S(P).SFCTCOMR instruction is shown below.

[Condition]
• The number of comments to be stored in the comment file: 1000
• Sequence steps in the SFC step in the SFC program: 1000 sequence steps
• The number of active steps: 40

Instruction	Condition			High Performance model QCPU		Process CPU	Redundant CPU
				QnCPU	QnHCPU		
S(P).SFCSCOMR	At instruction execution			280μs	120μs	120μs	120μs
	At END processing (read 1 comment)			780μs	350μs	350μs	350μs
S(P).SFCTCOMR	At instruction execution			300μs	130μs	120μs	120μs
	At END processing (read 1 comment)	• Transition condition for serial transition • Transition condition after selection branching		2.5ms	1.1ms	1.1ms	1.1ms
		• Transition condition after parallel coupling	Number of parallel couplings: 2	4.5ms	2.0ms	2.0ms	2.0ms
			Number of parallel couplings: 32	60.5ms ＊1	26.2ms	26.2ms	26.2ms

＊1: Indicates that the sequence steps in SFC steps consist of 800 sequence steps.

Instruction	Condition	Universal model QCPU			
		Q03UD(E)HCPU		Q04UD(E)HCPU, Q06UD(E)HCPU, Q10UD(E)HCPU, Q13UD(E)HCPU, Q20UD(E)HCPU, Q26UD(E)HCPU, Q50UDEHCPU, Q100UDEHCPU	
		Min.	Max.	Min.	Max.
S(P).SFCSCOMR	At instruction execution	190μs	193μs	176μs	177μs
S(P).SFCTCOMR		190μs	193μs	176μs	177μs

Instruction	Condition	Universal model QCPU			
		Q03UDVCPU		Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, Q26UDVCPU	
		Min.	Max.	Min.	Max.
S(P).SFCSCOMR	At instruction execution	54.6μs	65.0μs	51.3μs	65.0μs
S(P).SFCTCOMR		54.8μs	65.3μs	51.7μs	65.3μs

Instruction	Condition			Universal model QCPU			
				Q03UD(E)HCPU		Q04UD(E)HCPU, Q06UD(E)HCPU, Q10UD(E)HCPU, Q13UD(E)HCPU, Q20UD(E)HCPU, Q26UD(E)HCPU, Q50UDEHCPU, Q100UDEHCPU	
				SRAM card	Flash card	SRAM card	Flash card
S(P).SFCSCOMR	At END processing (read 1 comment)			3.3ms	4.5ms	2.5ms	4.0ms
S(P).SFCTCOMR	At END processing (read 1 comment)	• Transition condition for serial transition		3.7ms	5.3ms	3.3ms	5.0ms
		• Transition condition after selection branching		3.2ms	4.9ms	2.9ms	4.4ms
		• Transition condition after parallel coupling	Number of parallel couplings: 2	4.0ms	5.7ms	3.6ms	5.1ms
			Number of parallel couplings: 32	18.7ms	21.0ms	13.8ms	14.0ms

Instruction	Condition			Universal model QCPU	
				Q03UDVCPU	Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, Q26UDVCPU
				Standard RAM	Standard RAM
S(P).SFCSCOMR	At instruction execution			0.26ms	0.25ms
S(P).SFCTCOMR	At END processing (read 1 comment)	• Transition condition for serial transition		0.66ms	0.44ms
		• Transition condition after selection branching		0.66ms	0.35ms
		• Transition condition after parallel coupling	Number of parallel couplings: 2	0.87ms	0.55ms
			Number of parallel couplings: 32	1.28ms	0.75ms

3 - 23

3.4 Calculating the SFC Program Capacity

In order to express the SFC diagram using instructions, the memory capacity shown below is required. The method for calculating the SFC program capacity and the number of steps when the SFC diagram is expressed by SFC dedicated instructions is described in this section.

(1) Method for calculating the SFC program capacity

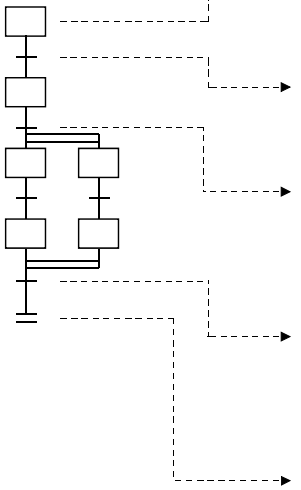
$$\text{SFC program capacity} = \underline{2} + \left(8 \times \underline{\text{max. created block No.}+1} \right) + \frac{\text{(block 0 capacity)+(block 1 capacity)+…+(block n capacity)}}{\text{Number of blocks being used}}$$

- $\underline{2}$: SFC program START (SFCP) and END (SFCPEND) instructions
- $\underline{\text{max. created block No.}+1}$: SFC file header capacity

$$\text{Capacity of blocks} = \underline{2} + \left(\underline{\text{number of steps where SFC diagram is expressed by SFC dedicated instructions}} \right) + \text{(operation output total for all steps)} + \text{(total number of transition conditions)}$$

- $\underline{2}$: Block START (BLOCK BLm) and END (BEND) instructions
- $\underline{\text{number of steps where SFC diagram is expressed by SFC dedicated instructions}}$: ∗ As shown below

∗ Number of steps where SFC diagram is expressed by SFC dedicated instructions



- Step (□, ⊟ ⊟)
  3 sequence steps (+) for step START (STEP ⬚ Sn) and END (SEND) instructions.
- Transition conditions (+)
  1) For serial transition or selective branching coupling
  4 sequence steps for transition START instruction (TRAN ⬚ TRn) and transition destination instruction (TSET Sn).
  2) For parallel branching
  Total number of steps for the transition START instruction (TRAN ⬚ TRn), and transition destination instructions (TSET Sn) for the number of parallel branches in question.
  3) For parallel coupling
  Total number of steps for the transition START instruction (TRAN ⬚ TRn), and the transition destination instructions (TSETSn) and coupling check instructions (TAND Sn) for the (number of parallel branchings in question−1.
- Jump (↳) , end step (⊥)
  Calculated as step 0 because it is included in the previous transition condition.
- Operation outputs for each step: The capacity per step is as follows
  - Total number of sequence steps for all instructions.
    (For details regarding the number of sequence steps for each instruction, refer to the QCPU Programming Manual (Common Instructions))
- Transition conditions: The capacity per transition condition is as follows
  - Total number of sequence steps for all instructions.
    (For details regarding the number of sequence steps for each instruction, refer to the QCPU Programming Manual (Common Instructions))

3 - 24

(2) Number of steps required for expressing the SFC diagram as SFC dedicated instructions
The following table shows the number of steps required for expressing the SFC diagram as SFC dedicated instructions.

Name	Ladder Expression	Number of Steps	Description	Required Number of Steps
SFCP START instruction	[SFCP]	1	Indicates the SFC program START	1 per program
SFCP END instruction	[SFCPEND]	1	Indicates the SFC program END	1 per program
Block START instruction	[BLOCK BLm]	1	Indicates the block START	1 per block
Block END instruction	[BEND]	1	Indicates the block END	1 per block
Step START instruction	[STEP ⬚ Si]	2	Indicates the step START (" ⬚ " varies according to the step attribute)	1 per step
Transition START instruction	[TRAN ⬚ TRj]	2	Indicates the transition START (" ⬚ " varies according to the step attribute)	1 per transition condition
Coupling check instruction	[TAND Si]	2	"Coupling completed" check occurs at parallel coupling	"[Number of parallel couplings] - [1]" per parallel coupling
Transition designation instruction	[TSET Si]	2	Designates the transition destination step	For serial transitions and selection transitions, 1 per transition condition; for parallel branching transitions, the number of steps is the same as the number of parallel couplings
Step END instruction	[SEND]	1	Indicates the step / transition END	1 per step

3 - 25

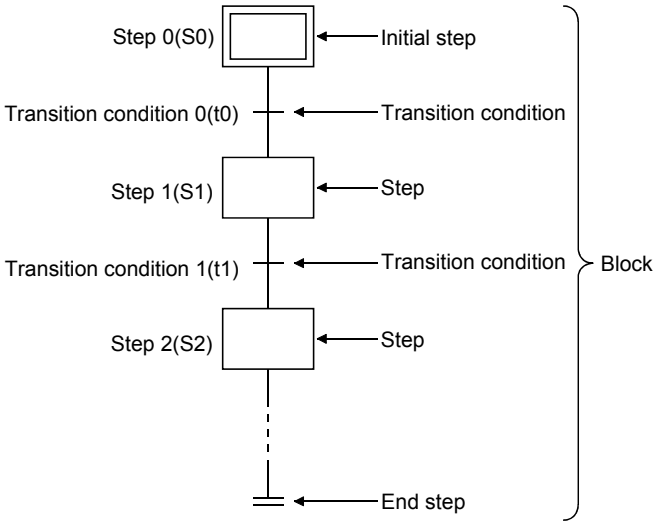
MEMO

3 - 26

4. SFC PROGRAM CONFIGURATION

This chapter explains the SFC program symbols, SFC control instructions and SFC information devices that comprise an SFC program.
When applying the program examples introduced in this manual to an actual system, ensure the applicability and confirm that it will not cause system control problems.

(1) As shown below, an SFC program consists of an initial step, transition conditions, intermediate steps, and an END step. The data beginning from the initial step and ending at the END step is referred to as a block.



(2) An SFC program starts at an initial step, executes a step following a transition condition in due order every time that transition condition is satisfied, and ends a series of operations at an end step.

(a) When the SFC program is started, the initial step is executed first.
While the initial step is being executed, whether the transition condition following the initial step (transition condition 0 (t0) in the figure) has been satisfied or not is checked.

(b) Only the initial step is executed until transition condition 0 (t0) is satisfied.
When transition condition 0 (t0) is satisfied, the execution of the initial step is stopped, and the step following the initial step (step 1 (S1) in the figure) is executed.
While step 1 (S1) is being executed, whether the transition condition following step 1 (transition condition 1 (t1) in the figure) has been satisfied or not is checked.

(c) When transition condition 1 (t1) is satisfied, the execution of step 1 (S1) is stopped, and the next step (step 2 (S2) in the figure) is executed.

(d) Every time the transition condition is satisfied in order, the next step is executed, and the block ends when the end step is executed.

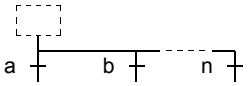
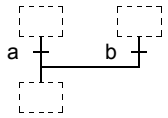
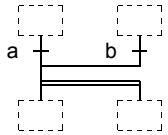
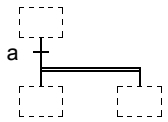
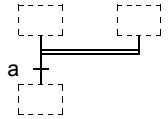
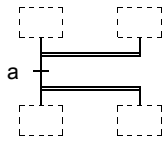
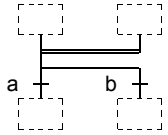
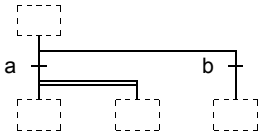
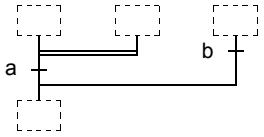
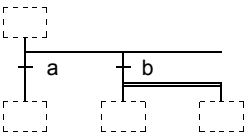
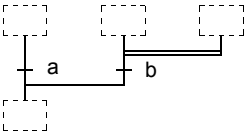
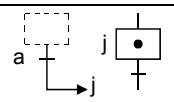
4.1 List of SFC Diagram Symbols

The symbols used in the SFC program are listed below.

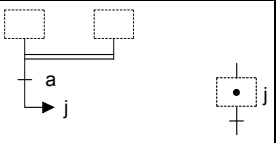
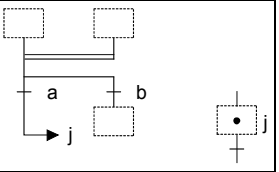
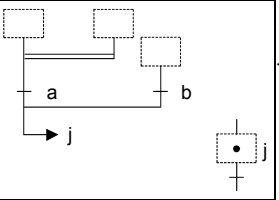
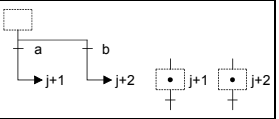
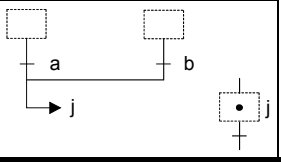
Class	Name		SFC Diagram Symbol	Remarks
Step	Initial step	When step No. is "0"	0 □	Any of these steps in 1 block *: Initial step at top left (column 1) of SFC diagram is fixed to No. 0. n = reset destination step No.
	Dummy initial step		0 ⊠	
	Coil HOLD initial step		0 SC	
	Operation HOLD step (without transition check) initial step		0 SE	
	Operation HOLD step (with transition check) initial step		0 ST	
	Reset initial step		0 Sn R	
	Initial step	When initial step No. is other than "0"	i □	Up to 31 steps in 1 block. i = step No. (1 to 511) n = reset destination step No.
	Dummy initial step		i ⊠	
	Coil HOLD initial step		i SC	
	Operation HOLD step (without transition check) initial step		i SE	
	Operation HOLD step (with transition check) initial step		i ST	
	Reset initial step		i Sn R	
	Step	Steps other than "initial" step	i □	Up to 512 steps in 1 block, including initial step (128 steps for Basic model QCPU) i = step No. (1 to 511) n = reset destination step No. m = movement destination block No.
	Dummy step		i ⊠	
	Coil HOLD step		i SC	
	Operation HOLD step (without transition check)		i SE	
	Operation HOLD step (with transition check)		i ST	
	Reset step		i Sn R	
	Block START step (with END check)		i Bm	
	Block START step (without END check)		i Bm	
	End step		⊥	More than one step can be used in 1 block.

4

4 - 2

Class	Name	SFC Diagram Symbol	Remarks
Transition	Serial transition		a, b = Transition condition No.
	Selection branching		
	Selection coupling		
	Selection coupling - parallel branching		
	Parallel branching		
	Parallel coupling		
	Parallel coupling - parallel branching		
	Parallel coupling - selection branching		
	Selection branching - parallel branching		
	Parallel coupling - selection coupling		
	Selection branching - parallel branching		
	Parallel coupling - selection coupling		
	Jump transition		a = Transition condition No. j = jump destination step No.

4 - 3

Class	Name	SFC Diagram Symbol	Remarks
Transition	End step transition		a, b = Transition condition No. j = jump destination step No.
	Selection coupling - Jump		
	Selection coupling - Selection branching - Jump		
	Selection coupling - Selection coupling - Jump		
	Selection branching - Jump		
	Selection coupling - Jump		

4 - 4

4.2 Steps

Steps are the basic units for comprising a block, and each step consists of operation outputs.

(1) The following table indicates the number of steps that can be used in one block.

CPU module type		Maximum number of steps in one block	Maximum number of steps for all blocks
Basic model QCPU		128 steps	1024 steps
High Performance model QCPU		512 steps	8192 steps
Process CPU			
Redundant CPU			
Universal model QCPU	Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU	128 steps	1024 steps
	Q03UDCPU, Q03UDVCPU, Q03UDECPU, Q04UDHCPU, Q04UDVCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDVCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDVCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDVCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	512 steps	16384 steps ＊1
LCPU	L02SCPU, L02SCPU-P, L02CPU, L02CPU-P	128 steps	1024 steps
	L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	512 steps	8192 steps

＊1: For the Universal model QCPU whose serial number (first five digits) is "12051" or earlier, the maximum number of steps for all blocks is 8192.

(2) Serial step numbers are assigned to the steps in creation order at the time of SFC program creation.
The user can specify the step numbers to change them within the range of the maximum number of steps in one block.
The step numbers are used for monitoring the executed step and for making a forced start or end with the SFC control instruction.

4 - 5

4.2.1 Step □ (without step attribute)

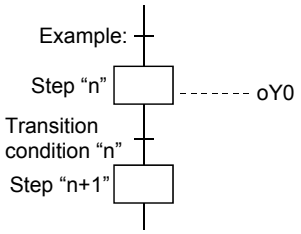
During processing of steps without attributes, the next transition condition is constantly monitored, with transition to the next step occurring when the condition is satisfied.

(1) The operation output status of each step (n) varies after a transition to the next step (n + 1), depending on the instruction used.

(a) When the OUT instruction is used (excluding OUT C ⬚)

When a transition to the next step occurs and the corresponding step becomes inactive, the output turned ON by the OUT instruction turns OFF automatically.

The timer also turns OFF its coil and contact and also clears its present value.



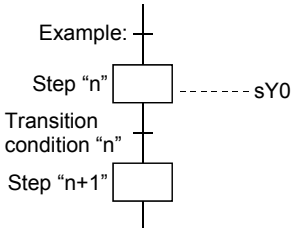
When transition condition “n” becomes satisfied at the step “n” operation output where Y0 is ON (in accordance with the OUT instruction), Y0 is automatically switched OFF.

4 - 6

(b) When the SET, basic or application instruction is used

If a transition to the next step occurs and the corresponding step becomes inactive, the device remains ON or the data stored in the device is held.

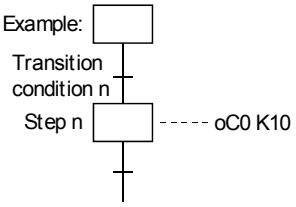
To turn OFF the ON device or clear the data stored in the device, use the RST instruction, etc. at another step.



When transition condition "n" becomes satisfied at the step "n" operation output where Y0 is ON (by SET instruction), the Y0 ON status will be maintained even after the transition to step "n + 1".

(c) When the oC ⬚ instruction is used:

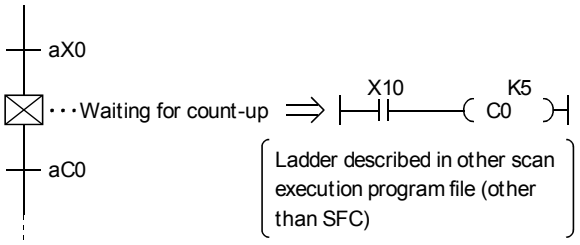
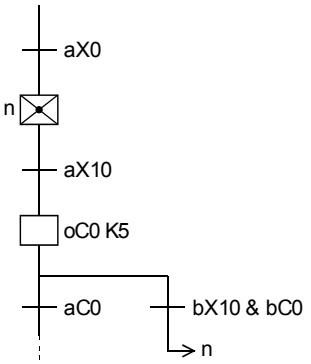
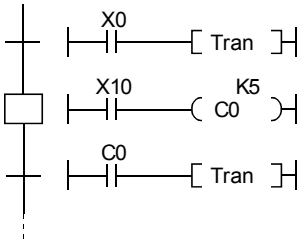
1) The counter counts once every time the transition condition is satisfied and the corresponding step that is inactive is activated.



Counter C0 counts once when transition condition n is satisfied and execution proceeds to step n.

2) To cause the counter to count once when the input condition turns ON and execution to proceed to the next step when the counter counts up,
   • Create a counter ladder in a sequence program; or
   • Create an SFC diagram using a jump transition on MELSAP-L.

   In the program example shown below, the counter counts once every time X10 turns ON, and execution proceeds to the next step when C0 counts up.

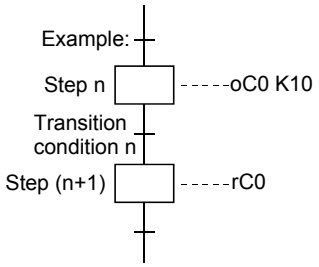
When creating counter ladder in sequence program	When using jump transition on MELSAP-L	When creating program with MELSAP3
		

4 - 7

3) When counter is reset

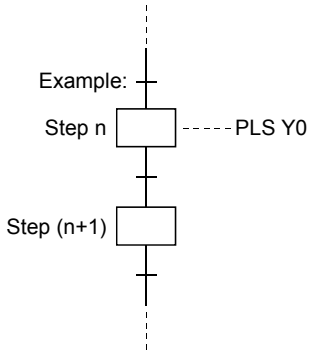
When a transition to the next step occurs before the reset instruction of the counter is executed, the present value of the counter and the ON status of the contact, which is ON, are held if the corresponding step becomes inactive.

To reset the counter, the RST instruction, etc. must be executed at another step.

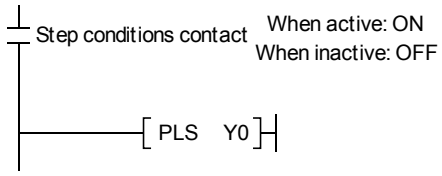


When counter C0 is reset at step (n+1) or later, the present value is cleared and the contact turns OFF.

(2) The PLS or P instruction used for the operation output of any step is executed every time the corresponding step turns from an inactive to an active status.



The program shown on the left is actually executed in a ladder as shown below. Because the step conditions contact is ON when the step is active and OFF when the step is inactive, the PLS or P instruction will be executed every time the corresponding step becomes active.



4 - 8

4.2.2 Initial step

The initial step represents the beginning of a block. Up to 32 initial steps per block can be designated.
When there are more than one initial step, the coupling enabled is only a selective coupling.
Execute the initial steps in the same way as executing the steps other than the initial step.

(1) Active steps at block START

When the block that has more than one initial step is started, the active steps change depending on the starting method as described below.

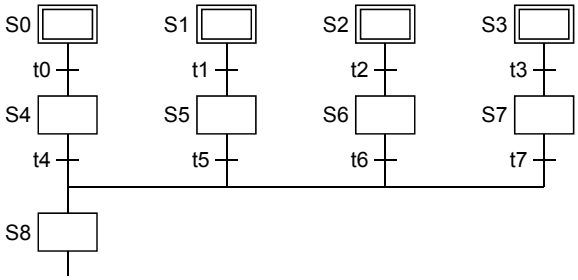
- When the block START step makes a start using (Bm, Bm)
- When a start is made using the block START instruction (sBLm) of the SFC control instructions
- When a forced start is made using the block START/END bit of the SFC information devices

→ All initial steps become active.

- When any of the initial steps is specified using the step control instruction (sBLm\Sn, sSn) of the SFC control instructions

→ Only the specified step becomes active.

(2) Transition processing performed when multiple initial steps become active



If steps are selectively coupled in the block that has more than one active initial steps, the step immediately after the coupling becomes active if any of the transition conditions immediately before the coupling is satisfied.
In the above program example, step 8 (S8) becomes active when any of transition conditions t4 to t7 is satisfied.
When, after the step immediately after the coupling (S8 in the above program example) becomes active, another transition condition immediately before the coupling (any of t4 to t7 in the above program example) is satisfied, reactivation processing is performed as a follow-up function.
The processing, which will be performed when another transition condition is satisfied with the step immediately after coupling being active, can be selected between STOP, WAIT and TRANSFER in the "Operation mode at transition to active step (double step START)" (refer to Section 4.7.6) in the block parameter setting of the SFC setting dialog box in the Tools menu.
For the Basic model QCPU, Universal model QCPU, and LCPU, the operation mode cannot be selected.
It operates in the default "TRANSFER" mode.

(3) The operation of the initial steps with step attributes is the same as that of the other steps.
Refer to Section 4.2.4 to Section 4.2.7.

4.2.3 Dummy step ☒

A dummy step is a waiting step, etc., which contains no operation output program.

(1) The transition condition following the corresponding step is always checked during execution of a dummy step, and execution proceeds to the next step when the transition condition is satisfied.

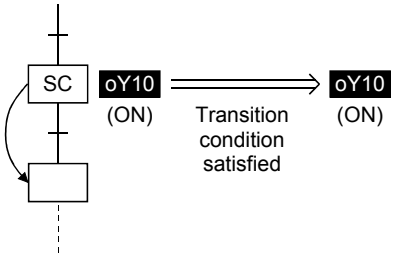
(2) The dummy step changes to a step (without step attribute, indication: □) when an operation output program is created.

4.2.4 Coil HOLD step SC

A coil HOLD step is a step where the coil output status is maintained in the transition to the next step. (The coil output is switched ON by the OUT instruction when the transition condition is satisfied.)

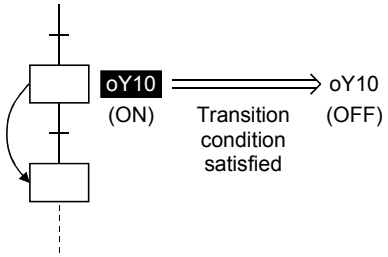
(1) During normal SFC program operation, the coil ON status (switched ON by OUT instruction when transition condition is satisfied) is automatically switched OFF before proceeding to the next step.
By designating an operation output step as a “coil HOLD step”, the coil ON status will remain in effect when proceeding to the next step.

When designated as a coil HOLD step



• At a designated coil HOLD step, “Y10” (switched ON by OUT instruction) will remain ON even when the transition condition is satisfied.

When not designated as a coil HOLD step



• At steps not designated as coil HOLD steps, “Y10” (switched ON by OUT instruction) is automatically switched OFF when the transition condition is satisfied.

(2) No ladder processing occurs following a transition to the next step.

4 - 10

(3) When a coil ON status (at coil HOLD step) has been maintained to the next step, the coil will be switched OFF at any of the following times:
(a) When the end step of the corresponding block is executed. (Except when SM327 is ON)
(b) When an SFC control instruction (rBLm) designates a forced END at the block in question.
(c) When an SFC control instruction (rBLm\Sn, rSn) designates a reset at the block in question.
(d) When a reset occurs at the device designated as the SFC information register's block START/END device.
(e) When a reset step for resetting the step in question becomes active.
(f) When the SFC START/STOP command (SM321) is switched OFF.
(g) When the coil in question is reset by the program.
(h) When the STOP instruction is executed with the stop-time output mode OFF.
(i) When S999 is designated at the reset step in the corresponding block.

(4) Block STOP processing
Make a block STOP using the STOP/RESTART bit of the SFC information devices or the block STOP instruction of the SFC control instructions.
The processing of the active step in the block where a block STOP was made is as described below.
(a) When the "block STOP-time operation output flag (SM325)" is OFF (coil output OFF)
• The step becomes inactive when the processing of the corresponding block is performed first after a block STOP request.
• All coil outputs turn OFF.
However, the coils turned ON by the SET instruction remain ON.
(b) When the "block STOP-time operation output flag (SM325)" is ON (coil output held)
The coil outputs remain ON during a block STOP and after a block RESTART.

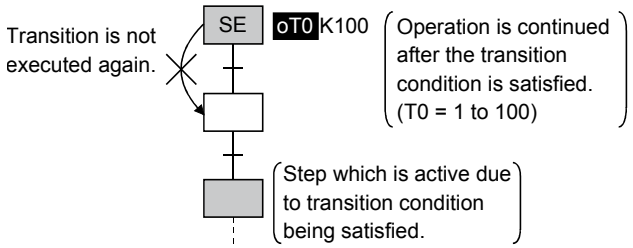
(5) Precautions when designating coil HOLD steps
(a) PLS instruction
When the execution condition of the PLS instruction is satisfied and the transition condition is satisfied at the same scan where the PLS instruction was executed, the device turned ON by the PLS instruction remains ON until the OFF condition in above (3) is satisfied.
(b) PLF instruction
When the execution condition of the PLF instruction is satisfied and the transition condition is satisfied at the same scan where the PLF instruction was executed, the device turned ON by the PLF instruction remains ON until the OFF condition in above (3) is satisfied.
(c) Counter
If the count input condition turns ON/OFF after a transition to the next step, the counter does not start counting.
(d) Timer
When a step transition occurs after the transition condition is satisfied with the coil of the timer ON, the timer stops timing and holds the then present value.

4.2.5 Operation HOLD step (without transition check) SE

An operation HOLD step (without transition check) is a step where the operation output processing of the corresponding step continues after a transition to the next step.
However, transition processing to the next step is not executed if the transition condition is satisfied again at the corresponding step.

(1) During normal SFC program operation, the coil ON status (switched ON by OUT instruction when transition condition is satisfied) is automatically switched OFF before proceeding to the next step.
When an operation output step is designated as an operation HOLD step (without transition check), the corresponding step will remain active after a transition to the next step, and operation output processing will continue.
After a transition, therefore, the timer continues timing.

(2) The transition conditions have been satisfied, so no transition condition check is performed after the next step becomes active.
Therefore, no step transition (subsequent transition) will occur even if the transition conditions for the relevant step are satisfied again.



(3) An operation HOLD step (without transition check) becomes inactive when any of the following occur:
(a) When the END step of the block in question is executed.
(b) When an SFC control instruction (rBLm) designates a forced END at the block in question.
(c) When the corresponding step is reset by the SFC control instruction (rBLm\Sn, rSn). (Except when SM327 is ON)
(d) When the device designated as the block START/END device of the SFC information devices is reset.
(e) When a reset step for resetting the step in question becomes active.
(f) When "S999" is designated at the reset step in the same block.
(g) When the SFC START/STOP command (SM321) is switched OFF.

(4) Block STOP processing

The following processing is performed when a block STOP request is issued to the corresponding block using the STOP/RESTART bit of the SFC information devices or the block STOP instruction of the SFC control instructions.

• STOP status timing

A STOP status is established after the block STOP request output occurs, and processing returns to the beginning of the block in question.

• Coil output

A coil output OFF or HOLD status will be established, depending on the output mode setting (see Section 4.7.3) at the time of the block STOP designated in the SFC operation mode. However, an ON status will be maintained for coil outputs which were switched ON by the SET instruction.

POINTS

(1) When the transition condition immediately before the corresponding step is satisfied or when the step is reactivated by a JUMP transition, a transition will occur again when the transition condition is satisfied.

(2) Double STARTs do not apply to reactivated steps.

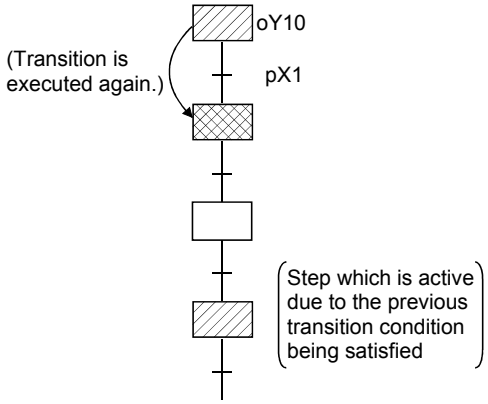
4 - 13

4.2.6 Operation HOLD step (with transition check) ST

An operation HOLD step (with transition check) is a step where the operation output processing of the corresponding step continues after a transition to the next step.
When the transition condition is satisfied again at the corresponding step, transition processing to the next step (reactivation) is executed.

(1) During normal SFC program operation, the coil ON status (switched ON by OUT instruction when transition condition is satisfied) is automatically switched OFF before proceeding to the next step.
When an operation output step is designated as an operation HOLD step (with transition check), the corresponding step will remain active after a transition to the next step, and operation output processing will continue.

(2) The transition condition will be checked after the transition condition is satisfied and the next step is activated.
Hence, when the transition condition of the corresponding step is satisfied again, a transition to the next step (subsequent transition) occurs to activate it.
At this time, the current step remains active.



POINTS

(1) Convert the transition conditions into pulses.
If they are not pulsed, transition processing to the next step is performed every scan while the condition is satisfied.

(2) When a double START occurs as the transition condition was satisfied with the transition destination step being active, the processing changes depending on the parameter setting.
The Basic model QCPU does not allow the parameters to be selected.
It operates in the default "Transfer" mode.
Refer to Section 4.7.6 for the parameter setting and the processing performed for each setting.

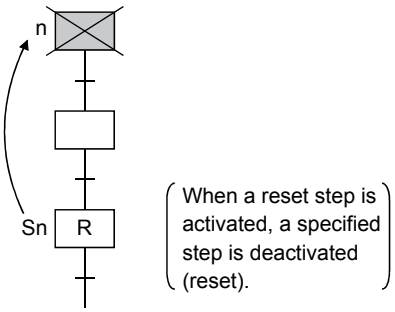
(3) The difference between the operation HOLD step (with transition check) and the operation HOLD step (without transition check) is whether the next step will be activated or not as a follow-up when the transition condition is satisfied again.

(3) An operation HOLD step (with transition check) becomes inactive when any of the following occur:
(a) When the end step of the corresponding block is executed.
(b) When an SFC control instruction (rBLm) designates a forced END at the block in question.
(c) When an SFC control instruction (rBLm\Sn, rSn) designates a reset at the block in question.
(d) When a reset occurs at the device designated as the SFC information register's block START/END device.
(e) When a reset step for resetting the step in question becomes active.
(f) When "S999" is designated at the reset step in the same block.
(g) When the SFC START/STOP command (SM321) is switched OFF.

(4) Block STOP processing
Make a block STOP using the STOP/RESTART bit of the SFC information devices or the block STOP instruction of the SFC control instructions.
The processing of the active step in the block where a block STOP was made is as described below.
(a) When the "block STOP-time operation output flag (SM325)" is OFF (coil output OFF)
The step becomes inactive when the processing of the corresponding block is performed first after a block STOP request.
• All coil outputs turn OFF.
• However, the coils turned ON by the SET instruction remain ON.
(b) When the "block STOP-time operation output flag (SM325)" is ON (coil output held)
The coil outputs remain ON during a block STOP and after a block RESTART.

4.2.7 Reset step R

A reset step is a step which designates a forced deactivation of another specified step (operation output).
The reset step deactivates the designated step in the current block before execution of the operation output every scan.
Except the deactivation of the specified step, the reset step execute the operation output with the same functions as a normal step (without step attributes).



(1) When deactivating only the designated step
Set the step number to be deactivated to the specified step number Sn.

(2) When deactivating all the held steps
Set "999" to the specified step number Sn.
When the number of the specified step is "999", all held steps of the coil HOLD steps, operation HOLD steps (without transition check) and operation HOLD steps (with transition check) in the current block are batch-deactivated.

POINT

(1) Only held steps can be deactivated by the reset step.
The following steps are not the targets of the reset step.
• HOLD steps that are active but not held
• Steps that are not specified as the HOLD steps

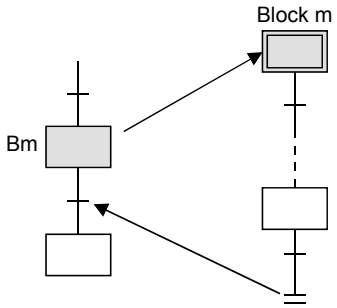
(2) For the Basic model QCPU, Universal model QCPU, and LCPU, a step of the CPU itself cannot be specified as a reset step.

4 - 16

4.2.8 Block START step (with END check)

A block START step (with END check) is the step where the specified block is started, and when the START destination block is then deactivated, the check of the transition condition to the next step is started.

(1) The operation of the block START step (with END check) is described below.
  (a) When activated, the block START step (with END check) starts the specified block.
  (b) No processing is performed until the START destination block is deactivated after its execution has ended.
  (c) When the START destination block is deactivated after its execution has ended, only the transition condition check is performed.
  (d) When the transition condition is satisfied, a transition to the next step occurs.



(2) A simultaneous start cannot be made for a single block.
  The block that has already started cannot be started, either.
  If either of the above starts is made, the following processing is performed depending on the setting of the operation mode at block double START. *1
  (Refer to Section 4.7.5 for details of the operation at block double START.)
  (a) When the setting of the operation mode at block double START is "STOP"
    A "BLOCK EXE. ERROR" (error code: 4620) occurs and the CPU module stops processing.
  (b) When the setting of the operation mode at block double START is the default setting of "WAIT"
    Processing is not performed and waits until the START destination block ends its execution.

POINTS
*1: For the Basic model QCPU, Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Universal model QCPU whose serial number (first five digits) is "12051" or earlier, and LCPU, the operation mode at double block START cannot be set. For these CPU modules, only the "WAIT" mode is available.

(3) A block START request can start multiple blocks simultaneously by performing a parallel transition (refer to Section 4.3.3).
  The steps in the simultaneously started blocks are processed in parallel.

(4) The following table indicates the number of steps that can be executed simultaneously in all blocks and the maximum number of active steps in a single block.

CPU module type		Number of steps that can be executed simultaneously in all blocks	Maximum number of active steps in one block
Basic mode QCPU		1024 steps	128 steps
High Performance model QCPU		1280 steps	256 steps
Process CPU			
Redundant CPU			
Universal model QCPU	Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU	1024 steps	128 steps
	Q03UDCPU, Q03UDVCPU, Q03UDECPU, Q04UDHCPU, Q04UDVCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDVCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDVCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDVCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	1280 steps	256 steps
LCPU	L02SCPU, L02SCPU-P, L02CPU, L02CPU-P	1024 steps	128 steps
	L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	1280 steps	256 steps

POINTS

(1) The block START step (with END check) cannot be described immediately before the coupling of a parallel coupling.
(The block START step (with END check) cannot be used for a wait.)
The block START step (without END check) can be described immediately before the coupling of a parallel coupling.

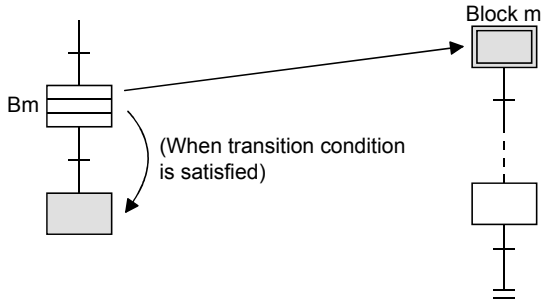
(2) The execution status of each block can be checked at another block using the block START/END bit (refer to Section 4.5.1) of the SFC information devices or the block activation check instruction (refer to Section 4.4.3) of the SFC control instructions.

4 - 18

4.2.9 Block START step (without END check)

A block START step (without END check) is the step where the specified block is started, and if the START destination block is active, the check of the transition condition to the next step is performed.

(1) The operation of the block START step (without END check) is described below.
  (a) When activated, the block START step (without END check) starts the specified block.
  (b) After starting the specified block, the step performs only the check of the transition condition.
  (c) When the transition condition is satisfied, execution proceeds to the next step without waiting for the START destination block to end.



(2) A simultaneous start cannot be made for a single block.
  The block that has already started cannot be started, either.
  If either of the above starts is made, the following processing is performed depending on the setting of the operation mode at block double START. *1
  (Refer to Section 4.7.5 for details of the operation at block double START.)
  (a) When the setting of the operation mode at block double START is "STOP"
    A "BLOCK EXE. ERROR" (error code: 4620) occurs and the CPU module stops processing.
  (b) When the setting of the operation mode at block double START is the default setting of "WAIT"
    Processing is not performed and waits until the START destination block ends its execution.

POINTS

*1: For the Basic model QCPU, Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Universal model QCPU whose serial number (first five digits) is "12051" or earlier, and LCPU, the operation mode at double block START cannot be set. For these CPU modules, only the "WAIT" mode is available.

(3) A block START request can start multiple blocks simultaneously by performing a parallel transition (refer to Section 4.3.3).
  The steps in the simultaneously started blocks are processed in parallel.

(4) The number of steps that can be executed simultaneously is a total of up to 1280 steps*2 for all blocks.
The number of steps that can be executed simultaneously in a single block is a maximum of 256 steps*3 including those of the HOLD steps.
*2: Up to 1024 steps for the following CPU modules:
- Basic model QCPU
- Universal model QCPU (Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU)
- LCPU (L02SCPU, L02SCPU-P, L02CPU, L02CPU-P)

*3: Up to 128 steps for the following CPU modules:
- Basic model QCPU
- Universal model QCPU (Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU)
- LCPU (L02SCPU, L02SCPU-P, L02CPU, L02CPU-P)

POINTS

The execution status of each block can be checked at another block using the block START/END bit (refer to Section 4.5.1) or the block activation check instruction (refer to Section 4.4.3) of the SFC control instructions.

4.2.10 End step

An end step indicates that a series of processings in the corresponding block is all ended.

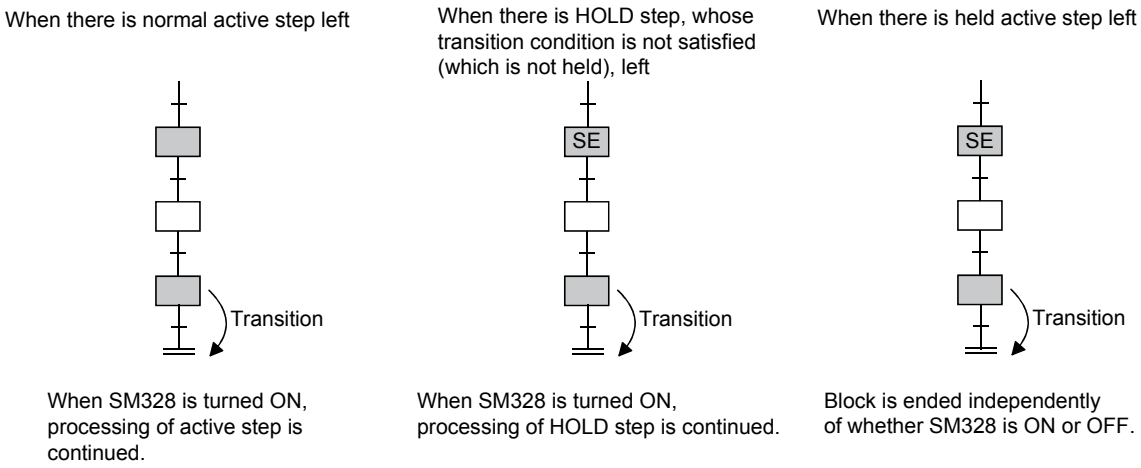
(1) When the end step is reached, the following processing is performed to end the block.
  (a) All steps in the block are deactivated.
    (The held step are also deactivated.)
  (b) The coil outputs turned ON by the OUT instruction are all turned OFF.
    When the special relay for output mode at end step execution (SM327) is ON, however, the coil outputs of the held steps all remain ON.

POINTS

(1) SM327 is valid only when the end step is reached.
  When a forced end is made by the block END instruction, etc., the coil outputs of all steps are turned OFF.

(2) SM327 is valid for only the HOLD steps being held.
  The outputs of the HOLD steps that are not held as the transition conditions are not satisfied are all turned OFF.

(2) When the special relay for clear processing mode at arrival at end step (SM328) is turned ON, the execution of the active step other than the one held in the block can be continued when the end step is reached. *1
  (The block is not ended if the end step is executed.)
  However, when there is only the held step left in the block at arrival at the end step, the held step is deactivated and the block ends if SM328 is ON.



REMARKS

*1: For the Basic model QCPU, Universal model QCPU, and LCPU, SM328 can be used to continue execution of active steps other than the one held in the block.

4 - 21

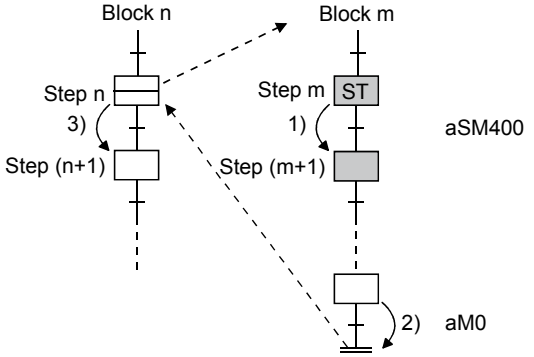
POINTS

The following gives the precautions to be taken when SM328 is turned ON.

(1) When there is only the held step left at arrival at the end step, that held step is deactivated if SM328 is ON.
When the user does not want to turn OFF the coil output of the held step suddenly, it can be prevented by turning ON SM327.

(2) If a block is started at the block START step when SM328 is ON, execution returns to the source as soon as there are no non-held active steps in the block.

(3) Do not describe an always satisfied transition condition immediately after the operation HOLD step (with transition check).



1) Since the transition condition is always satisfied, step (m+1) remains an active step (non-held active status).
2) If M0 turns ON and the transition condition is satisfied, block m cannot be ended.
3) Since block m is not ended, execution cannot proceed to step (n+1).

(a) When the transition condition immediately after the operation HOLD step (with transition check) is always satisfied, the next step is kept in a "non-held active status". Therefore, the block cannot be ended when SM328 is ON.
Further, if this block has been started at the block START step (with END check), processing cannot be returned to the START source step.

(b) When it is desired to describe an always satisfied transition condition immediately after the operation HOLD step (with transition check), make provision so that the block can be forcibly ended from outside.

(3) After end step execution, a restart is performed as described below.

Block No.		Restarting Method
Block 0	START condition of block 0 is set to "Auto START ON" in the SFC setting of the PLC parameter dialog box	• Execution automatically returns to the initial step again, and processing is executed repeatedly.
	START condition of block 0 is set to "Auto START OFF" in the SFC setting of the PLC parameter dialog box	• A restart is made when any of the following is executed. 1) When another START request is received from another block (when the block START step is activated) 2) When the block START instruction of the SFC control instructions is executed 3) When the block START/END bit of the block information devices is forcibly turned ON
All blocks other than block 0		

4 - 22

4.2.11 Instructions that cannot be used with operation outputs

Table 4.1 indicates the instructions that cannot be used with operation outputs.

Table 4.1 Unusable Instruction List

Class	Instruction Symbol	Symbol	Function	Remarks
Master control	MC	MC N □ No.1_D	Master control set	
	MCR	MCR N □	Master control reset	
End	FEND	FEND	Main routine program end	
	END	END	Sequence program end	
Program branch	CJ	CJ P □	Conditional jump	Label P cannot be used, either.
	SCJ	SCJ P □	Delayed jump	
	JMP	JMP P □	Unconditional jump	
	GOEND	GOEND	Jump to END	
Program control	IRET	IRET	Return from interrupt program	Label I cannot be used, either.
Structuring	BREAK	BREAK Ⓓ P □	Repetitive forced end	
	RET	RET	Return from subroutine	
Debugging troubleshooting	CHKST *1	CHKST	CHK instruction start	
	CHK *1	CHK	Specific format error check	
	CHKCIR *1	CHKCIR	Check pattern change start	
	CHKEND *1	CHKEND	Check pattern change end	
SFC dedicated instruction	SFCP	SFCP	SFC program start	
	SFCPEND	SFCPEND	SFC program end	
	BLOCK	BLOCK Ⓢ	SFC block start	
	BEND	BEND	SFC block end	
	STEP? (? = N, D, SC, SE, ST, R, C, G, I, ID, ISC, ISE, IST, IR)	STEP? Ⓢ	SFC step start	
	TRAN? (? = L, O, OA, OC, OCA, A, C, CA, CO, COC)	TRAN? Ⓢ	SFC transition start	
	TAND	TAND Ⓢ	SFC coupling check	
	TSET	TSET Ⓢ	SFC transition destination designation	
	SEND	SEND	SFC step end	

*1: The Basic model QCPU, Universal model QCPU, and LCPU do not support the instruction.

4 - 23

4.3 Transition

A transition is the basic unit for comprising a block, and is used by specifying a transition condition. A transition condition is a condition for execution to proceed to the next step, and execution proceeds to the next step when the condition is satisfied.

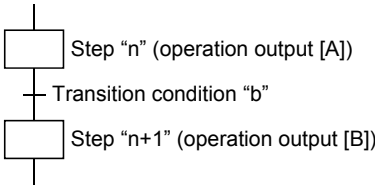
Table 4.2 Transition Condition Type List

Type	Function Outline
Serial transition	• When the transition condition is satisfied, execution proceeds from the current step to the subsequent step.
Selection transition (branch/coupling)	• A single step branches out into multiple transition conditions. • Among those multiple transition conditions, execution proceeds to only the step in the line where the transition condition is satisfied first.
Parallel transition (branch/coupling)	• Execution simultaneously proceeds to all multiple steps that branch from a single step. • When all steps immediately before a coupling are activated, execution proceeds to the next step when the common transition condition is satisfied.
Jump transition	• When the transition condition is satisfied, execution proceeds to the specified step in the same block.

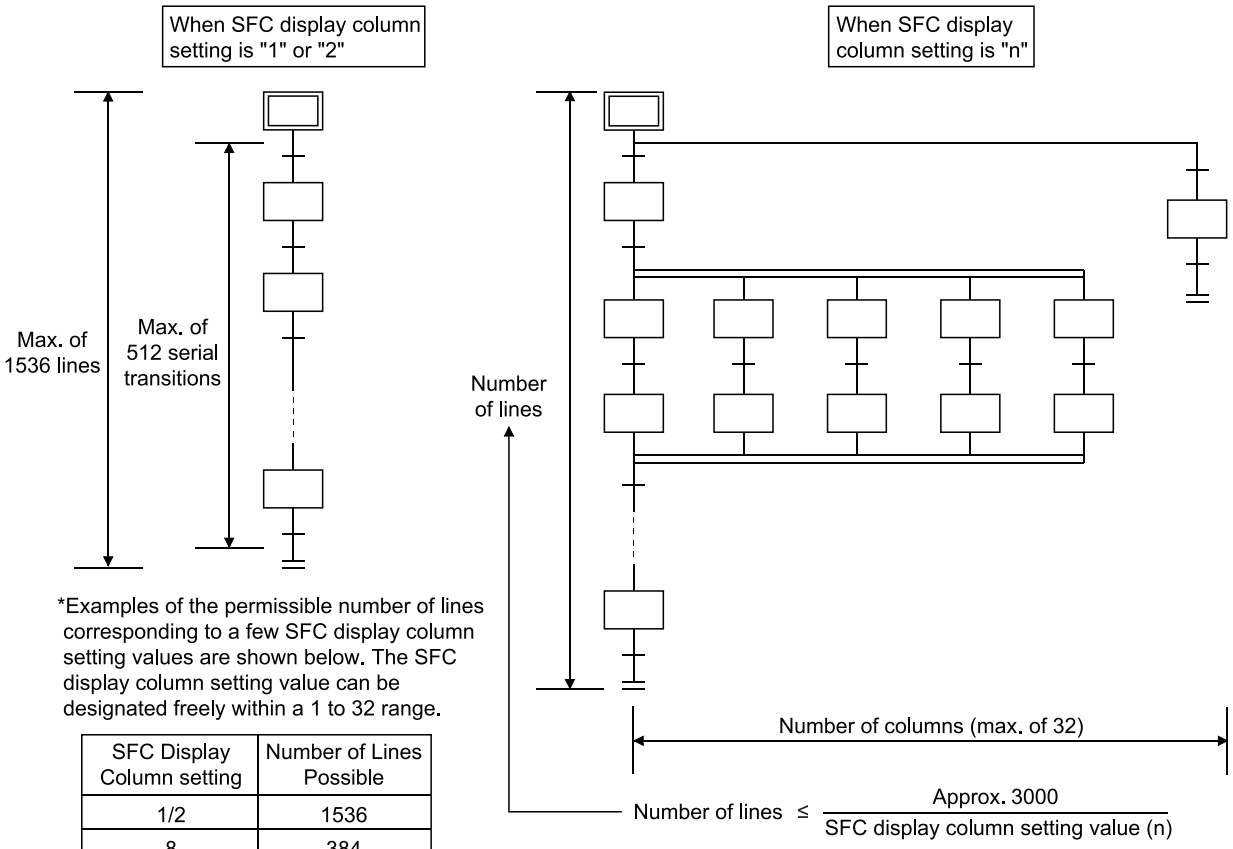
4 - 24

4.3.1 Serial transition

"Serial transition" is the transition format in which processing proceeds to the step immediately below the current step when the transition condition is satisfied.

	• When transition condition "b" becomes satisfied at step "n" (operation output [A]) execution, operation output [A] will be deactivated, and processing will proceed to step "n+1" (operation output [B]).

(1) A maximum of 512*1 serial transition steps (□, ☐, ⊥) can be described in each block.
Therefore, a maximum of 512* serial transitions (+) can be described.
However, there is a restriction on the number of lines as indicated below depending on the SFC display column setting.
*1: 128 for the Basic model QCPU, Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, L02SCPU, L02SCPU-P, L02CPU and L02CPU-P

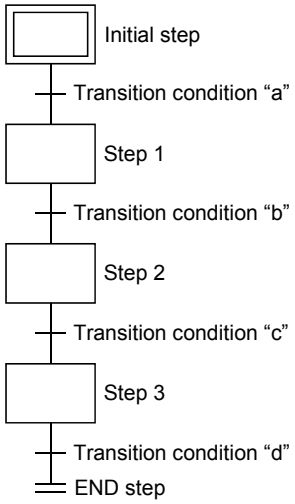


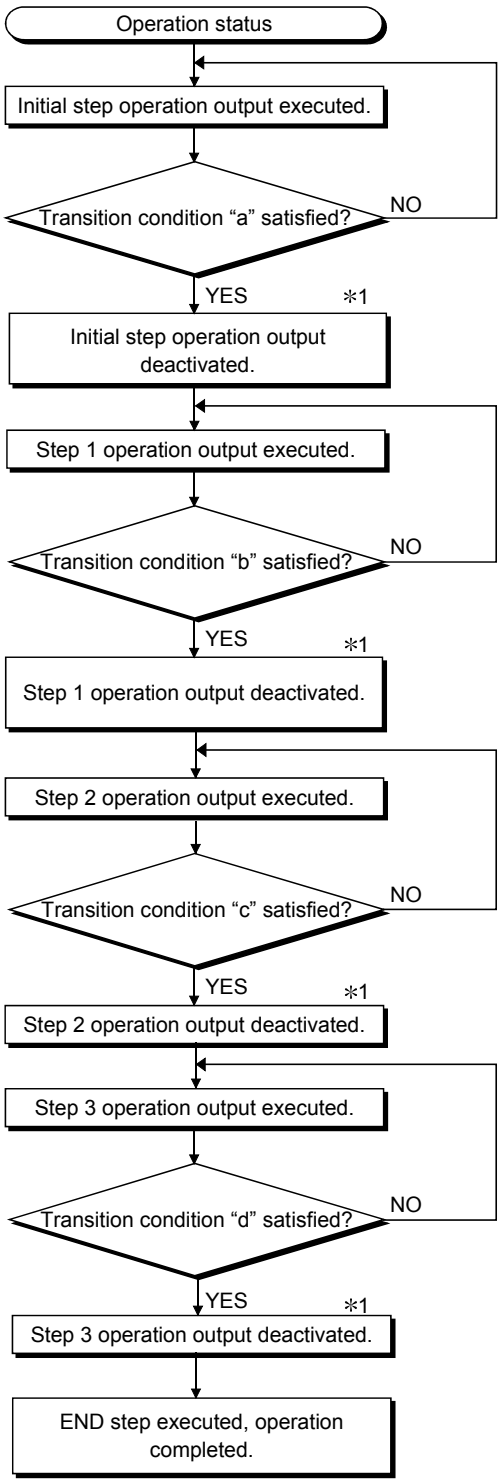
*Examples of the permissible number of lines corresponding to a few SFC display column setting values are shown below. The SFC display column setting value can be designated freely within a 1 to 32 range.

SFC Display Column setting	Number of Lines Possible
1/2	1536
8	384
16	192
22	138
28	108
32	96

4 - 25

(2) Serial transition operation flowchart



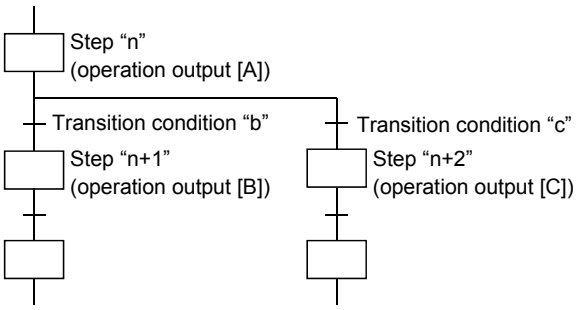
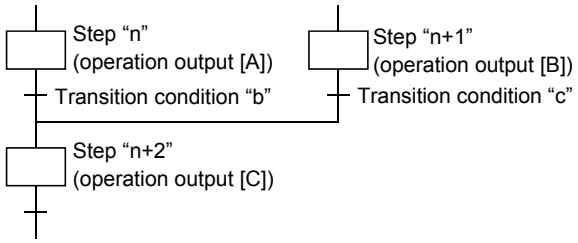


∗1 For steps with attribute designations, processing occurs in accordance with the attributes.

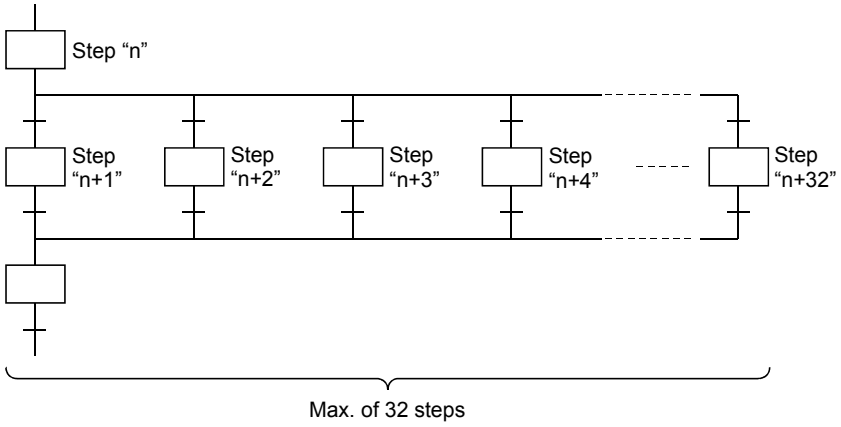
4 - 26

4.3.2 Selection transition

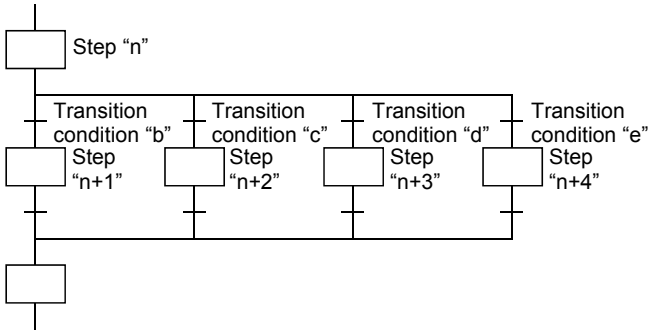
A “selection transition” is the transition format in which several steps are coupled in a parallel manner, with processing occurring only at the step where the transition condition is satisfied first.

Branch		• From step “n”, processing will proceed to either step “n+1” or step “n+2”, depending on which transition condition (“b” or “c”) is satisfied first. • If both transition conditions are satisfied simultaneously, the condition to the left will take precedence. Step “n” will then be deactivated. • Subsequent processing will proceed from step to step in the selected column until another parallel coupling selection occurs.
Coupling		• When the transition condition (“b” or “c”) at the executed branch is satisfied, the executed step ([A] or [B]) will be deactivated, and processing will proceed to step “n+2”.

(1) Up to 32 steps can be available for selection in the selection transition format.



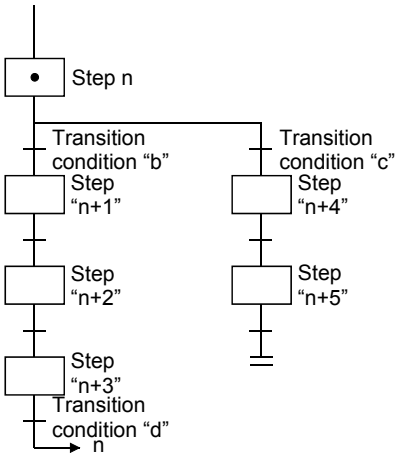
(2) When two or more selection step transition conditions are satisfied simultaneously, the left-most condition will take precedence.



Example: If transition conditions “c” and “d” are satisfied simultaneously, the step “n+2” operation output will be executed.

4 - 27

(3) In a selection transition, a coupling can be omitted by a jump transition or end transition.



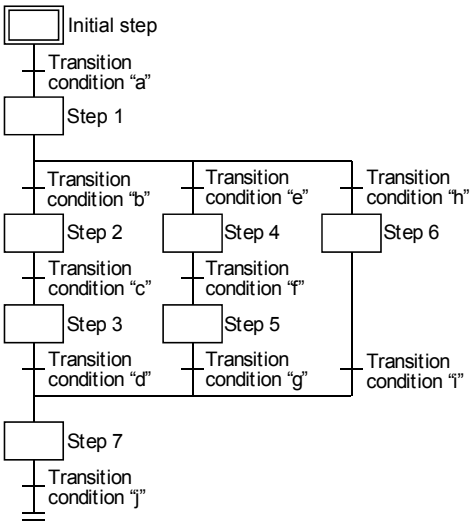
When transition condition "b" is satisfied at the step "n" operation output, processing will proceed in order through steps "n+1", "n+2" and "n+3". When transition condition "d" is satisfied, processing will jump to step "n". (For details on "jump transitions", see Section 4.3.4.)

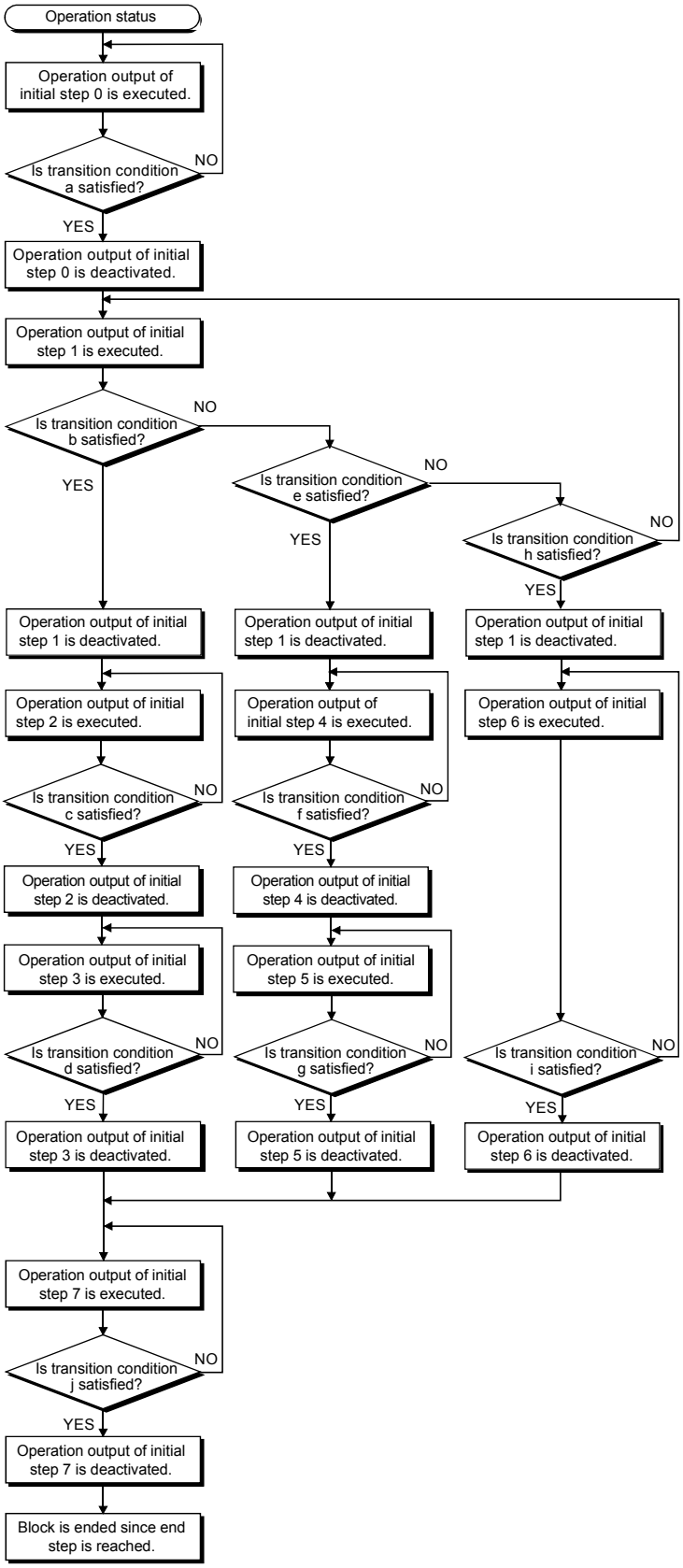
POINTS

In a selective transition, the number of branches and the number of couplings may be different. However, a selection branch and parallel coupling or a parallel branch and selection coupling cannot be combined.

4 - 28

(4) Selection transition operation flowchart

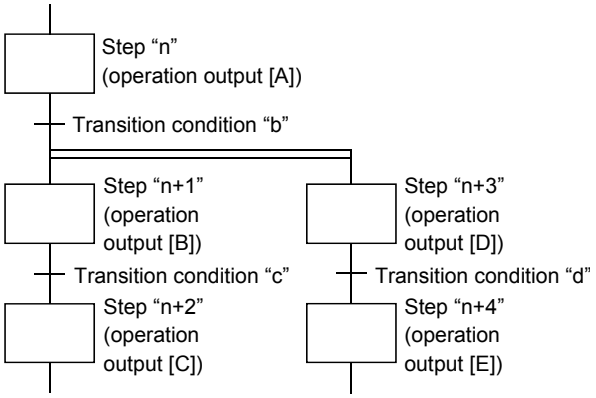
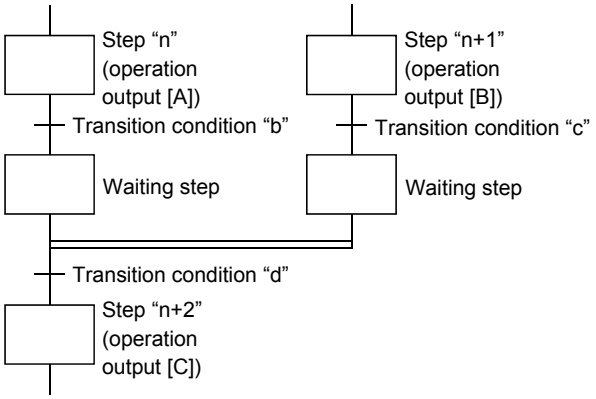




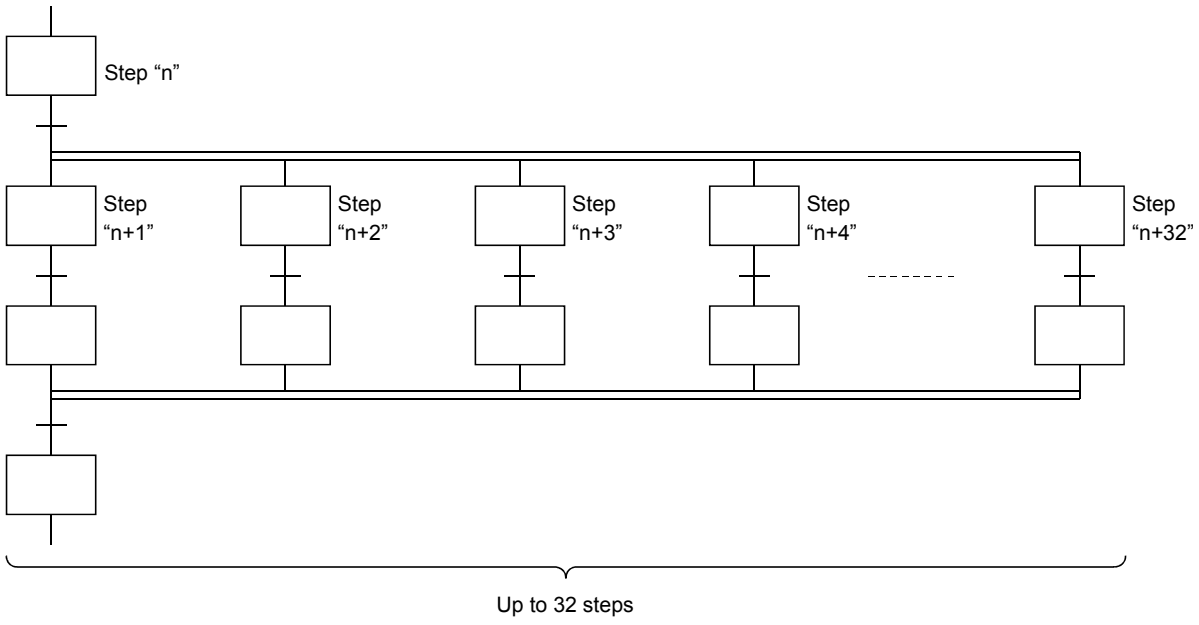
4 - 29

4.3.3 Parallel transition

"Parallel transition" is the transition format in which several steps linked in parallel are processed simultaneously when the relevant transition condition is satisfied.

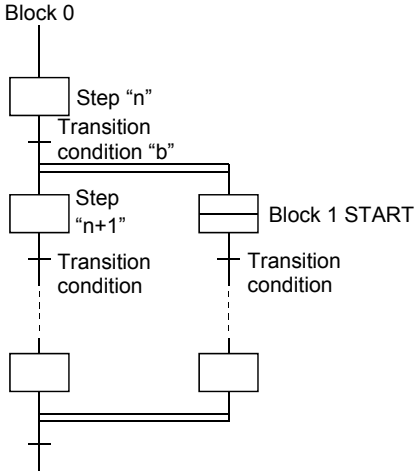
Branch		• From step "n", processing will proceed simultaneously to steps "n+1" and "n+3" when transition condition "b" is satisfied. • Processing will proceed to step "n+4" when transition condition "c" is satisfied, and to step "n+4" when transition condition "d" is satisfied.
Coupling		• When transition conditions "b" and "c" are satisfied at step "n" and step "n+1" execution, steps "n" and "n+1" will be deactivated, and processing will proceed to the waiting steps. • Waiting steps are used to synchronize parallel processing operations. Parallel processing steps always proceed to a waiting step. When condition "d" is satisfied at the waiting steps, processing will proceed to step "n+2". • Waiting steps are dummy steps which require no operation output ladder.

(1) Up to 32 steps can processed simultaneously with the parallel transition format.



4 - 30

(2) If another block is started by the parallel processing operation, the START source block and START destination block will be executed simultaneously. (In the example below, processing from step "n+1" will be executed simultaneously with block 1.)



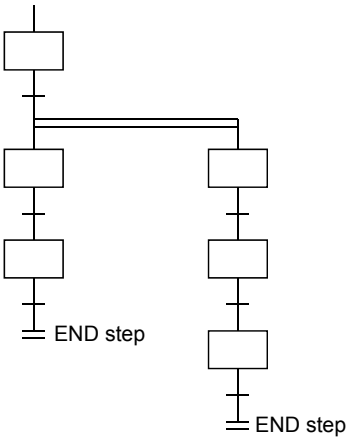
When condition "b" is satisfied at step "n" execution, processing will proceed to step "n+1" and block 1 will be started. Blocks "0" and "1" will then be processed simultaneously.

(3) The following table indicates the number of steps that can be executed simultaneously in all blocks and the maximum number of active steps in a single block.
If the number of simultaneously processed steps exceeds the value in the following table, an error occurs and the CPU module stops processing.

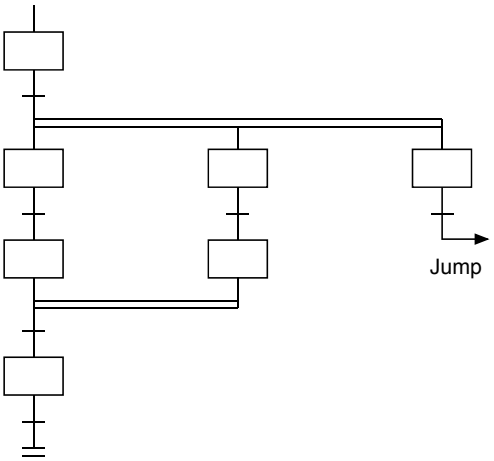
CPU module type		Number of steps that are processed simultaneously	Maximum number of active steps in one block
Basic mode QCPU		1024 steps	128 steps
High Performance model QCPU		1280 steps	256 steps
Process CPU			
Redundant CPU			
Universal model QCPU	Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU	1024 steps	128 steps
	Q03UDCPU, Q03UDVCPU, Q03UDECPU, Q04UDHCPU, Q04UDVCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDVCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDVCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDVCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU	1280 steps	256 steps
LCPU	L02SCPU, L02SCPU-P, L02CPU, L02CPU-P	1024 steps	128 steps
	L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	1280 steps	256 steps

4 - 31

(4) Couplings must be provided when the parallel transition format is used. Program creation is impossible without couplings.
Example: Program without couplings (Cannot be designated)

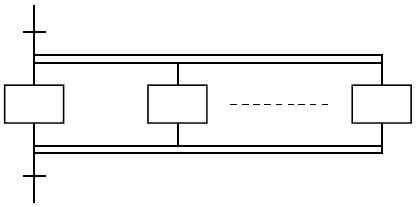


Each column ends
at the END step.



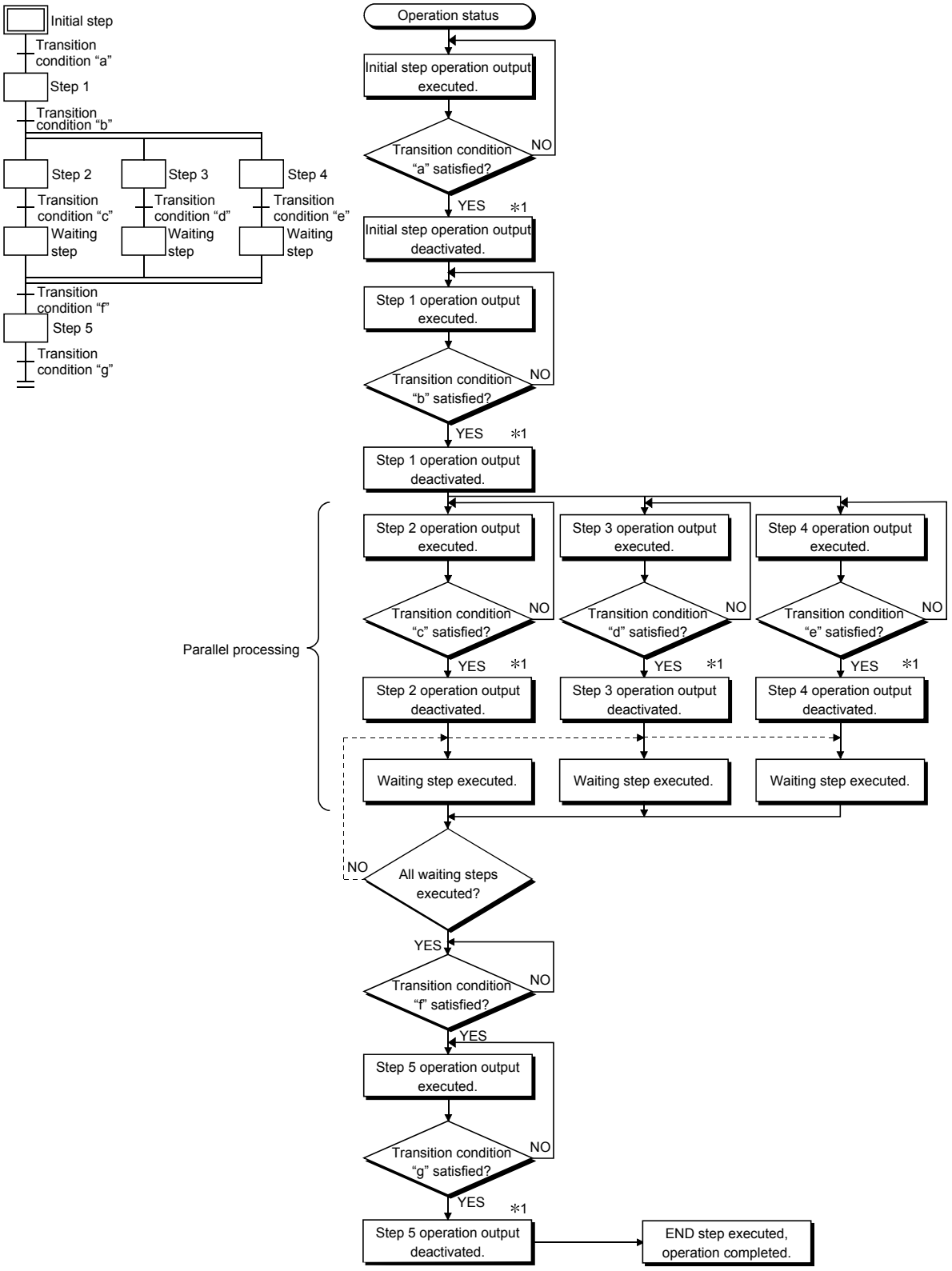
Jump transition (see Section 4.3.4)
occurs without coupling

(5) As a rule, a waiting step must be created prior to the coupling.
However, in cases such as the example below where each of the parallel transition columns consist of only 1 step (program without a transition condition between the parallel transition branch and the coupling), a waiting step is not required.



4 - 32

(6) Parallel transition operation flowchart

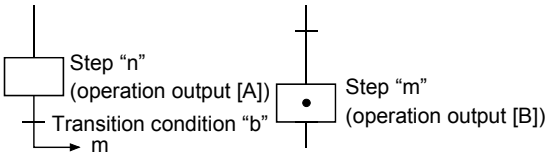


*1 For steps with attribute designations, processing occurs in accordance with the attributes.

4 - 33

4.3.4 Jump transition

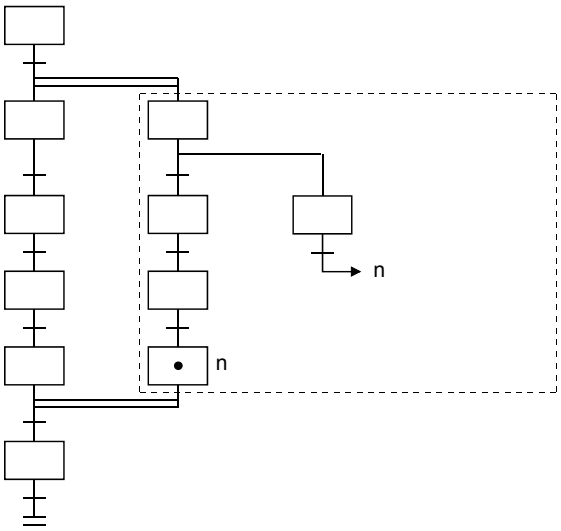
A “jump transition” is a jump to a specified step within the same block which occurs when the transition condition is satisfied.

	• When condition “b” is satisfied at step “n” execution, step “n” (operation output [A]) is deactivated, and processing proceeds to step “m”.

(1) There are no restrictions regarding the number of jump transitions within a single block.

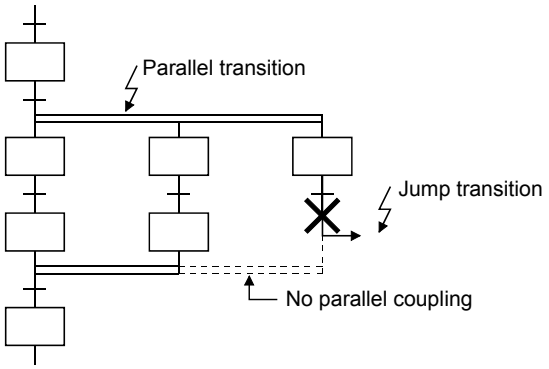
(2) In the parallel transition format, only jumps in the vertical direction are possible at each of the branches.

Example 1: Jump transition program in vertical direction from branch to coupling

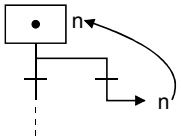


A program of a jump transition to another vertically branched ladder, a jump transition for exiting from a parallel branch, or a jump transition to a parallel branch from outside a parallel branch cannot be created.

Example 2: Program for exiting from parallel branch (cannot be designated)



(3) Do not specify a jump transition to the current step when the transition condition is satisfied as shown below.



4 - 34

4.3.5 Precautions for creating operation output (step)/transition condition programs

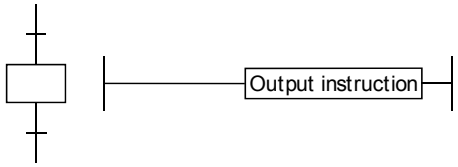
This section explains the precautions for creating operation output (step) and transition condition programs.

(1) Step program

(a) Step program expression method

Since a step program cannot use contacts and instructions equivalent to contacts, the operation output of an active step is executed when the transition condition is satisfied. Also, a step attribute is used to hold the output or continue operation when the step is activated.

A step program is expressed as a ladder circuit as shown below.



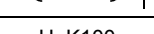
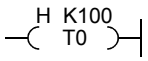
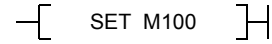
REMARKS

If a program is not created at a step, an error will not occur.
In this case, no processing is performed until the transition condition immediately following the corresponding step is satisfied.

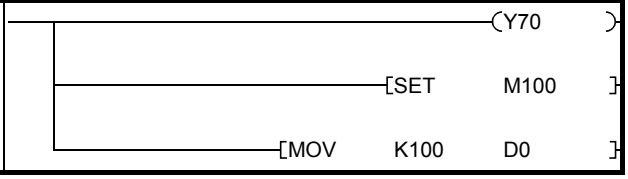
4 - 35

(2) MELSAP-L program description

The MELSAP-L describes step programs in the following format:

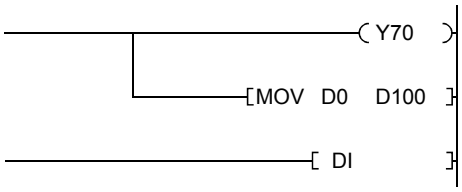
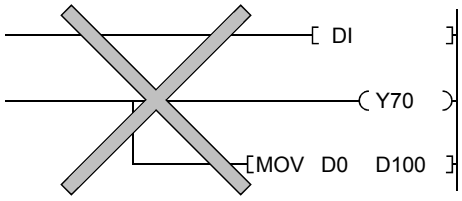
Instruction	Example of MELSAP-L format	Example of list format	Example of circuit symbol
Output (o)	oY70	OUT Y70	
High speed timer (h)	hT0 K100	OUTH T0 K100	
Set (s)	sM100	SET M100	
Reset (r)	rM200	RST M200	
The instructions other than above (o, h, s, r) are the same as the list.	MOV K100 D0	MOV K100 D0	

To arrange several operation outputs in parallel in the same step, separate them with "," (comma).

MELSAP-L format	Circuit
oY70, sM100, MOV K100 D0	

(3) Precautions on description

(a) Describe the instructions which do not require execution conditions (e.g. DI or EI) at the last of each operation output.
With the MELSAP-L, execution conditions like contacts cannot be created as the operation output. Therefore, the instruction which requires execution conditions cannot be created after the instruction which exists individually and does not require execution conditions.

	MELSAP-L description	Circuit expression
Example of proper description	oY70, MOV D0 D100, DI	
Example of improper description	DI, oY70, MOV D0 D100	

(b) Up to 24 instructions can be described for one operation output.

POINT

- The timer continues measurement while the step is active.
- The instruction which requires execution conditions cannot be created between FOR and NEXT instructions.

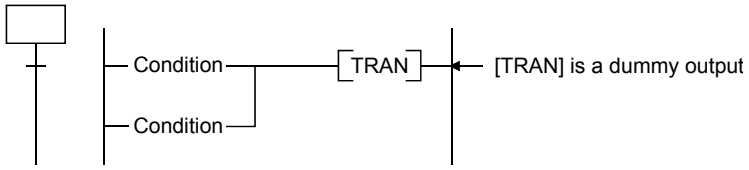
4 - 36

(4) Transition condition program

(a) Transition condition program expression

Transition condition programs can be used only for contact or contact-equivalent instructions.

The transition condition program is expressed as the following ladder circuit:



(b) Instructions used

Instructions which can be used in a transition condition program are listed below.

Class	Instruction Code	Instruction expression	Function	CPU Module Type		
				Basic model QCPU	High Performance Model QCPU, Process CPU, Redundant CPU	Universal model QCPU, LCPU
Contacts	LD AND OR	a Ⓢ & a Ⓢ | a Ⓢ	Operation START (N/O contact) Serial connection (N/O contact) Parallel connection (N/O contact)	○	○	○
	LDI ANI ORI	b Ⓢ & b Ⓢ | b Ⓢ	Operation START (N/C contact) Serial connection (N/C contact) Parallel connection (N/C contact)	○	○	○
Contacts	LDP ANDP ORP	p Ⓢ & p Ⓢ | p Ⓢ	Leading edge pulse operation START Leading edge pulse serial connection Leading edge pulse parallel connection	○	○	○
	LDF ANDF ORF	f Ⓢ & f Ⓢ | f Ⓢ	Trailing edge pulse operation START Trailing edge pulse serial connection Trailing edge pulse parallel connection	○	○	○
	INV	& INV	Operation result inversion	○	○	○
	MEP MEF	& MEP & MEF	Operation results converted to leading edge pulse (step memory) Operation results converted to trailing edge pulse (step memory)	○	○	○
	EGP EGF	& EGP Ⓓ & EGF Ⓓ	Operation results converted to leading edge pulse (memory) Operation results converted to trailing edge pulse (memory)	○	○	○

○: Usable, ×: Unusable

4 - 37

Class	Instruction Code	Instruction expression	Function	CPU Module Type		
				Basic model QCPU	High Performance Model QCPU, Process CPU, Redundant CPU	Universal model QCPU, LCPU
Comparison operation	LD □ AND □ OR □	□ ⓢ1 ⓢ2 & □ ⓢ1 ⓢ2 | □ ⓢ1 ⓢ2 □ (=, <>, >, >=, <, <=)	BIN16 bit data comparison	○	○	○
	LDD □ ANDD □ ORD □	D □ ⓢ1 ⓢ2 & D □ ⓢ1 ⓢ2 | D □ ⓢ1 ⓢ2 □ (=, <>, >, >=, <, <=)	BIN32 bit data comparison	○	○	○
	LDE □ ANDE □ ORE □	E □ ⓢ1 ⓢ2 & E □ ⓢ1 ⓢ2 | E □ ⓢ1 ⓢ2 □ (=, <>, >, >=, <, <=)	Floating decimal point data comparison	×	×	○
	LD$ □ AND$ □ OR$ □	$ □ ⓢ1 ⓢ2 & $ □ ⓢ1 ⓢ2 | $ □ ⓢ1 ⓢ2 □ (=, <>, >, >=, <, <=)	Character string data comparison	×	○	○

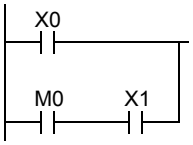
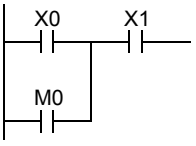
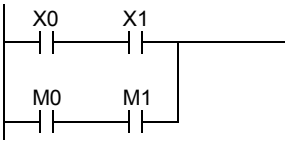
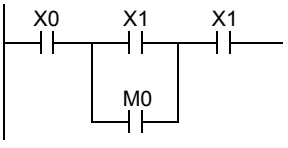
○: Usable, ×: Unusable

(5) Serial and parallel connections being mixed

When serial and parallel connections exist in the same transition condition, a serial connection "&" has priority.

Use "( )" to give a parallel connection " | " higher priority.

Example)

Example of serial and parallel connections being mixed	Ladder example
aX0 | aM0 & aX1	
(aX0 | aM0) & aX1	
aX0 & aX1 | aM0 & aM1	
aX0 & (aX1 | aM0) & aM1	

4 - 38

4.4 Controlling SFC Programs by Instructions (SFC Control Instructions)

SFC control instructions can be used to check a block or step operation status (active/inactive), or to execute a forced START or END, etc.
Using the SFC control instructions with an SFC program created with SFC program symbols, the SFC program can be usually controlled easily. (A program execution management SFC program cannot be controlled by using SFC control instructions.)
The types and functions of the SFC control instructions will be explained.

Name	Instruction Expression	Function	CPU Module Type		
			Basic model QCPU	High Performance Model QCPU, Process CPU, Redundant CPU	Universal model QCPU, LCPU
Step operation status check instruction 0	[a, &a, Ia, b, &b, Ib] Sn ＊1	• Checks a specified step in a specified block to determine if the step is active or inactive.	○	○	○
	[a, &a, Ia, b, &b, Ib] BLm\Sn				
Forced transition check instruction	[a, &a, Ia, b, &b, Ib] TRn ＊1	• Checks a specified step in a specified block to determine if the transition condition (by transition control instruction) for that step was satisfied forcibly or not.	×	○	×
	[a, &a, Ia, b, &b, Ib] BLn\TRn				
Block operation status check instruction	[a, &a, Ia, b, &b, Ib] BLm	• Checks a specified block to determine if it is active or inactive.	○	○	○
Active steps batch readout instruction	MOV(P) K4Sn Ⓓ ＊1	• Active steps in a specified block are read to a specified device as bit information.	○	○	○
	MOV(P) BLm\K4Sn Ⓓ				
	DMOV(P) K8Sn Ⓓ ＊1				
	DMOV(P) BLm\K8Sn Ⓓ				
	BMOV(P) K4Sn Ⓓ Kn ＊1				
	BMOV(P) BLm\K4Sn Ⓓ Kn				
Block START instruction	s BLm	• A specified block is forcibly started (activated) independently and is executed from an initial step.	○	○	○
Block END instruction	r BLm	• A specified block is forcibly ended (deactivated).	○	○	○
Block STOP instruction	PAUSE BLm	• A specified block is temporarily stopped.	○	○	○
Block restart instruction	RSTART BLm	• The temporary stop status at a specified block is canceled, with operation resuming from the STOP step.	○	○	○
Step START instruction	s Sn ＊1	• A specified block is forcibly started (activated) independently and is executed from a specified step.	○	○	○
	s BLm\Sn				
Step END instruction	r Sn ＊1	• A specified step in a specified block is forcibly ended (deactivated).	○	○	○
	r BLm/Sn				
	SCHG Ⓓ ＊2	• The instruction execution step is deactivated, and a specified step is activated.	×	○	×
Transition control instruction	s TRn ＊1	• A specified transition condition at a specified block is forcibly satisfied.	×	○	×
	s BLm\TRn				
	r TRn ＊1	• The forced transition at a specified transition condition in a specified block is canceled.	×	○	×
	r BLm\TRn				
Block switching instruction	BRSET Ⓢ	• Blocks subject to the "*1" SFC control instruction are designated.	×	○	○＊3

○: Usable, ×: Unusable

4 - 39

*1: In a sequence program, block 0 is the instruction execution target block.
In an SFC program, the current block is the instruction execution target block.
The instruction execution target block can be changed with the block switching instruction (BRSET).
Note, however, that the following CPU modules cannot use the BRSET instruction.
• Basic model QCPU
• Universal model QCPU whose serial number (first five digits) is "13101" or earlier
• LCPU

*2: Can be used at the step of an SFC program.
An error occurs if it is executed in a sequence program other than an SFC program.

*3: The Universal model QCPU whose serial number (first five digits) is "13102" or later can execute this instruction.

POINTS

(1) Either of the following errors occurs if the SFC control instruction is executed from the sequence program when the special relay for SFC program start/stop (SM321) is OFF.
• Instruction that specifies a block: BLOCK EXE. ERROR (error No.: 4621)
• Instruction that specifies a step: STEP EXE. ERROR (error No.: 4631)

(2) The SFC block (BL) and step relay (S) in the Basic model QCPU, High Performance model QCPU, Process CPU, Redundant CPU, and Universal model QCPU (except for High-speed Universal model QCPU) cannot be index-modified.

(3) The SFC block (BL) and step relay (S) in the High-speed Universal model QCPU can be index-modified within the following range.
• BL0 to BL319 for the SFC block (BL)
• Range that is set in the Device tab of the PLC parameter dialog box for the step relay (S)
Note that the range will be S0 to S511 when the step relay (S) in SFC blocks is index-modified.

(4) Do not use the SFC control instructions in interrupt programs or fixed scan execution type programs.
If used, operation of the SFC program cannot be guaranteed.

(5) The step relay (S) can be used only in the following instructions.
• Step activation check instruction
• Active step batch read instruction
• Step START instruction
• Step END instruction

(6) The following instructions are described as follows when used in the format other than the MELSAP-L.
• s....SET
• r....RST

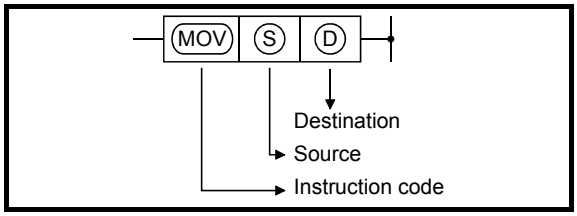
POINT

Beginning from Section 4.4.1 of this manual, the following table is used in the explanations of the various instructions. The table contents are explained below.

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J ☐\☐		Intelligent Function Module U☐\G☐	Index Z☐	Constant K, H	Expansion SFC BLm\Sn	Other		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓢ	—							—	○	○	BIN16/ BIN32	○	○	—	—	○	—
Ⓓ	○							—	—	—	BIN16/ BIN32						

1) 2) 3) 4) 5)

1) Ladder symbols are indicated in this area.



Destination .....Data destination following the operation.
Source .....Where data is stored prior to the operation.

2) Usable devices are indicated at this area.
- Devices indicated by a circle mark (O) can be used with the instruction in question. The device application classifications are shown below.

4 - 41

Device Class	Internal (System, User)		File Register R	Link Direct J □\□		Intelligent Function Module U□\G□	Index Z □	Expansion SFC	Constant	Other
	Bit	Word		Bit	Word					
Usable devices	FX, FY, S, SM, X, Y,M, L, F, V,B, T, C, SB	A, VD, SD, T, C, D, W, SW, FD, ST	R, ZR	J □ \X J □ \Y J □ \B J □ \SB	J □ \W J □ \SW	U □ \G	Z	BLm\Sn BLm\Trm	Decimal hexadecimal real number constant character string constant	P, I, J, U, DX, DY, N, BL, TR, BL\S

• When a device name is indicated in the “constant”, “expansion SFC”, or the “other” column, only that device may be used.
Example:
If “K, H” is indicated in the “constant” column, only a decimal (K) or hexadecimal (H) constant may be used.
Real number constants (E) and character string constants ($) may not be used.

3) The data type for the designated device is indicated here.

• Bit	Indicates a bit data operation.	
• BIN16	Indicates 16-bit binary value processing.	1 word used.
• BIN32	Indicates 16-bit binary value processing.	2 words used.
• Character string	Indicates character string processing.	Variable number of words.
• Device Indicates.....	device name and first device processing.	Variable number of words.

4) The type of program which can be used with the instruction in question is indicated here.

5) The request destination for the instruction in question is indicated here.

4 - 42

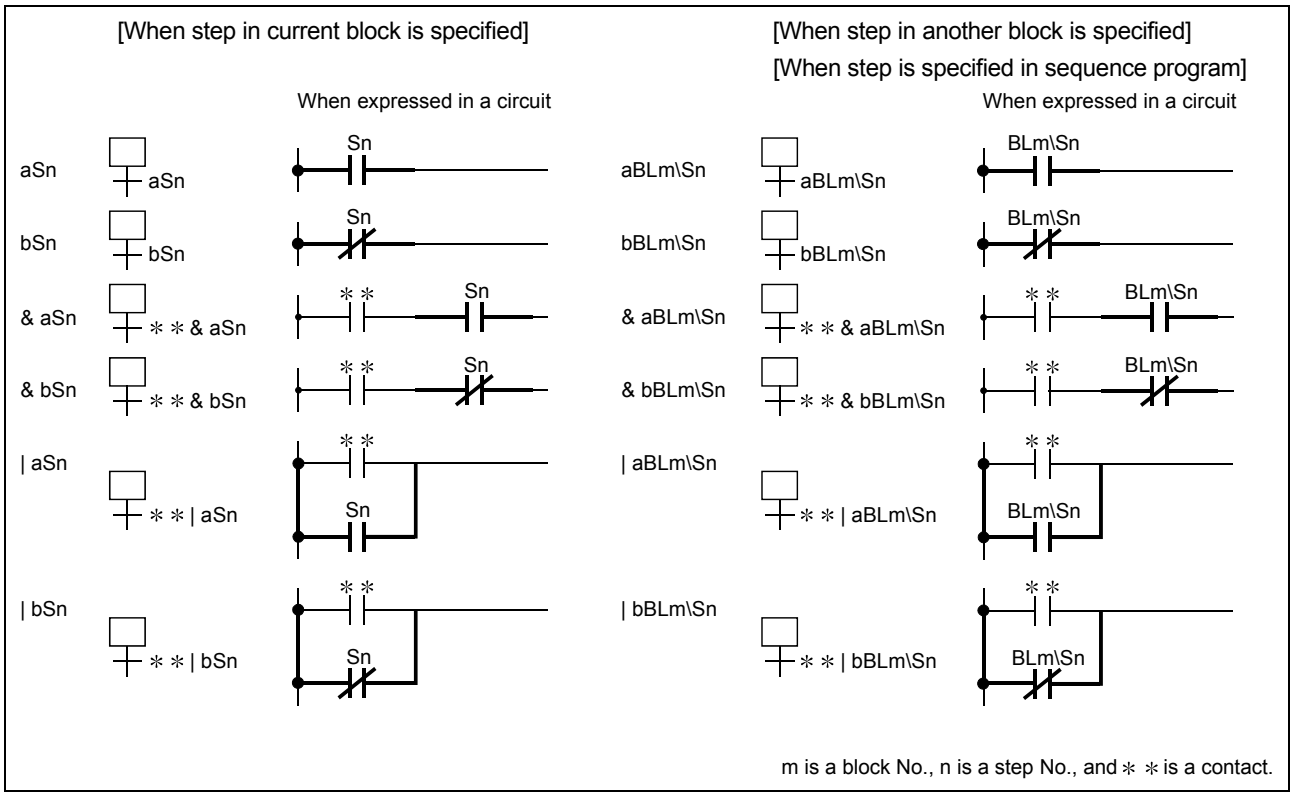
Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	△*	○	○	○	○	○

* : The serial number (first five digits) shall be 04122 or later.

4.4.1 Step operation status check instructions (a, b, &a, &b, la, lb)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J □\□		Intelligent Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn	Other		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓢ	⊛		—					—	○	—	Device name	○	—	○	—	○	—

⊛ : Only step relay (S) can be used



m is a block No., n is a step No., and ＊＊ is a contact.

[Functions]

(1) Checks a specified step in a specified block to determine if the step is active or inactive.

(2) The contact status changes as described below depending on whether the specified step is inactive or active.

	Contact of N/O Contact Instruction	Contact of N/C Contact Instruction
Inactive	OFF	ON
Active	ON	OFF

4 - 43

(3) Specify the step as described below.

(a) In the case of SFC program

1) Use "Sn" when specifying the step in the current block.

2) Use "BLm\Sn" when specifying the step in another block in the SFC program.

(b) In the case of sequence program

1) Use "BLm\Sn" when executing the step activation check instruction.

2) When the block number is not specified, specify the block number with the BRSET instruction.

Note, however, that the following CPU modules cannot use the BRSET instruction. When no block number is specified, the block 0 is set.

• Basic model QCPU

• Universal model QCPU whose serial number (first five digits) is "13101" or earlier

• LCPU

(4) If the step does not exist in the SFC program is specified, the contact remains OFF.

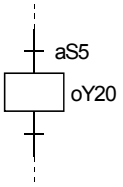
REMARKS

As the “Sn” device is treated as a virtual device, the contact on the monitor of a peripheral device does not turn ON/OFF. If the internal device is ON, the coil instruction is switched ON for operations.

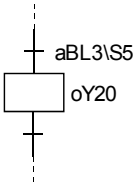
[Program Examples]

(1) The following program checks the status of step 5 in block 3 and turns ON Y20 when step 5 becomes active.

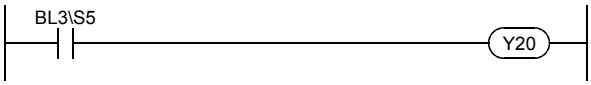
When step is designated by operation output of block 3



When step is designated by operation output of other than block 3

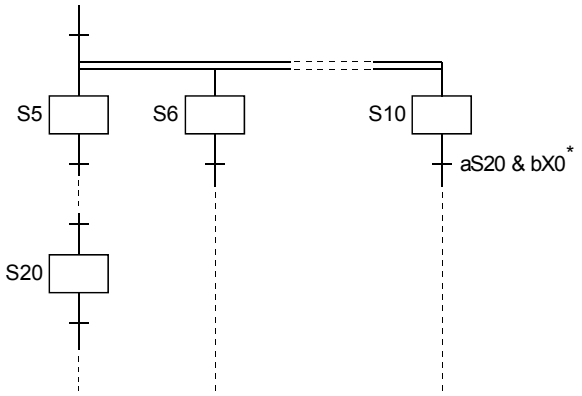


When step is designated by sequence program



4 - 44

(2) The following program executes a step synchronously with another step of a parallel branch.



Related Instructions

1) SFC control instructions
- Block switching instruction (BRSET) ..... See Section 4.4.11.
- Step control instruction (SCHG) ..... See Section 4.4.10.
- Active step batch readout instruction
  (MOV(P), DMOV(P), BMOV(P)) ..... See Sections 4.4.4 and 4.4.5.

REMARKS

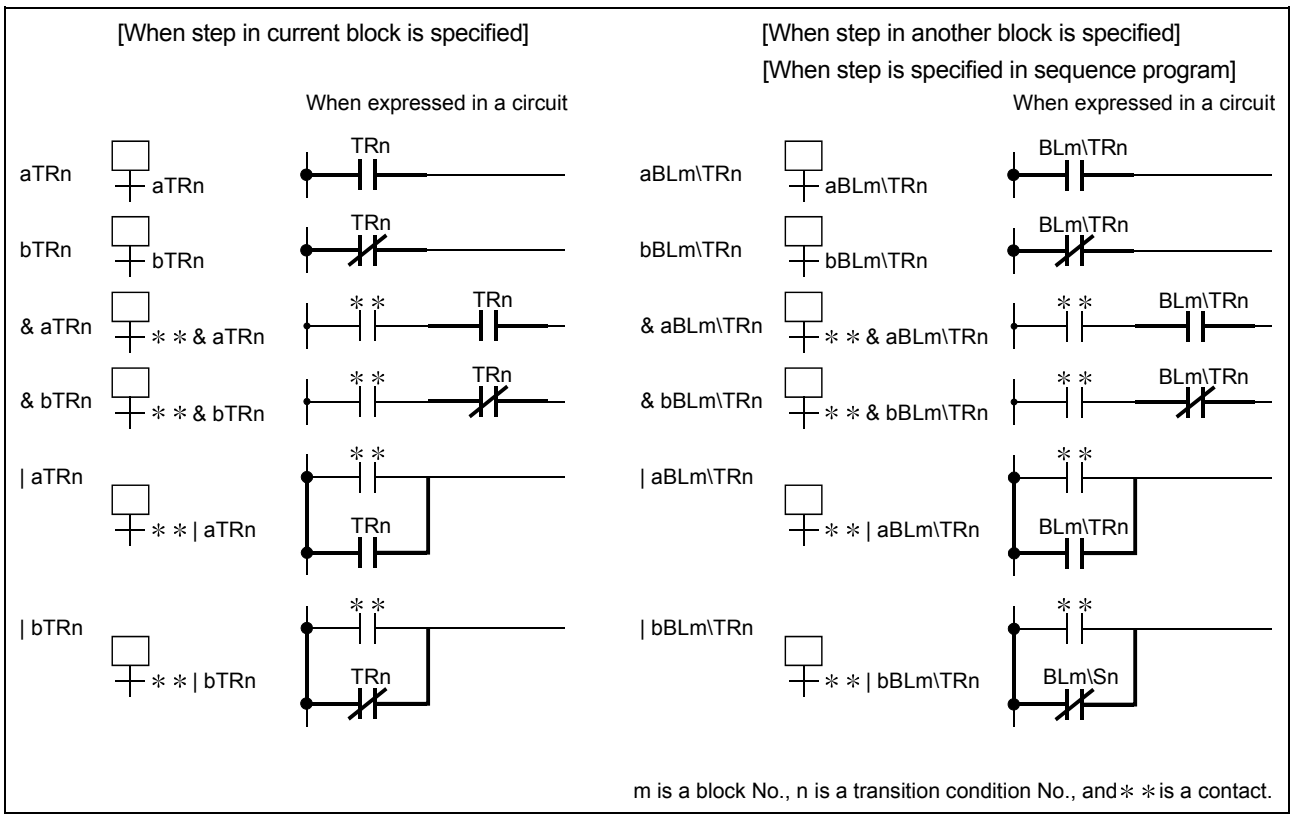
*: Indicates that X0 is used as a user interlock condition.

4 - 45

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	×	○	×	○	○	×

4.4.2 Forced transition check instruction (a, b, &a, &b, la, lb)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J ☐\☐		Intelligent Function Module U☐\G☐	Index Z☐	Constant K, H	Expansion SFC BLm\TRn	Other TRn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓢ	–							–	○	○	Device name	○	–	○	–	–	○



m is a block No., n is a transition condition No., and＊＊is a contact.

[Function]

(1) Checks whether or not the specified transition condition of the specified block is specified for forced transition by the forced transition EXECUTE instruction (SET BLm\TRn).

(2) The contact status changes as described below depending on whether the specified transition condition is specified for a forced transition or not.

	Contact of N/O Contact Instruction	Contact of N/C Contact Instruction
When specified for forced transition	ON	OFF
When not specified for forced transition	OFF	ON

4 - 46

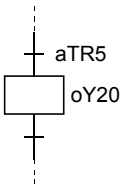
(3) Specify the transition as described below.
(a) In the case of SFC program
1) Use "Sn" when specifying the step in the current block.
2) Use "BLm\Sn" when specifying the step in another block in the SFC program.
(b) In the case of sequence program
1) Use "BLm\Sn" when executing the step activation check instruction.
2) When the block number is not specified, specify the block number with the BRSET instruction.

(4) If the transition condition in question does not exist in the SFC program, it will remain OFF.

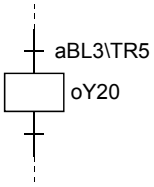
[Program Examples]

(1) The following program turns ON Y20 when transition condition 5 of block 3 is specified for a forced transition.

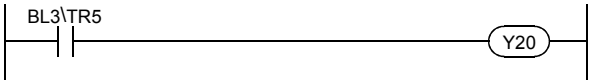
When step is designated by operation output of block 3



When step is designated by operation output of other than block 3



When step is designated by sequence program



Related Instructions

1) SFC control instructions
• Transition control instructions
(sTRn, sBLm\TRn,
rTRn, rBLm\TRn).....See Section 4.4.9.
• Block switching instruction (BRSET).....See Section 4.4.11.

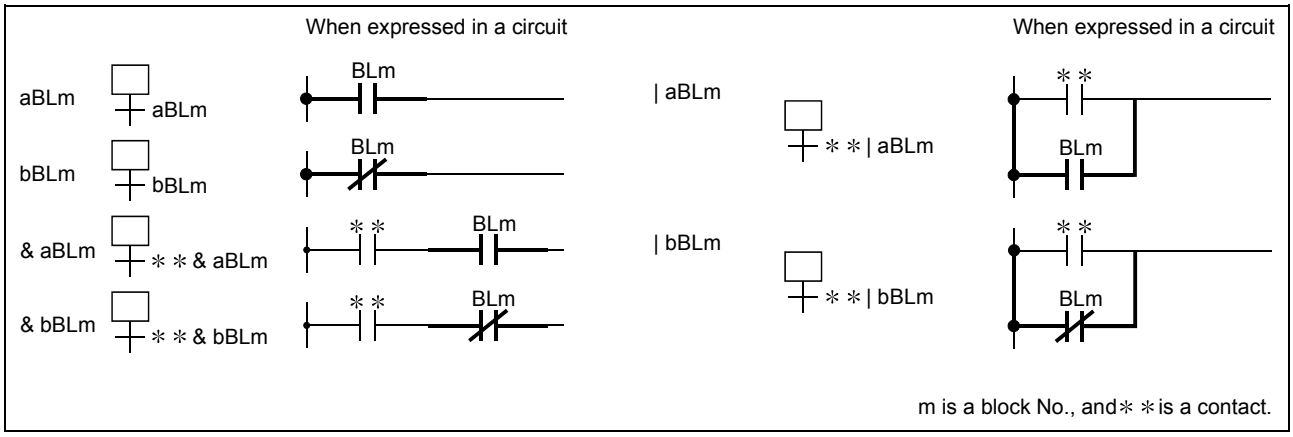
POINTS
This instruction checks, from the first sequence step of the specified block in series, whether or not the specified transition condition number is existed. Because of this, processing time of the instruction differs depending on the program capacity of the specified block (number of sequence steps), a maximum of hundred and several tens ms may be taken. In case of occurring WDT error (error code: 5001), change the WDT setting value with the PLC RAS setting in the PLC parameter.

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	△*	○	○	○	○	○

* : The serial number (first five digits) shall be 04122 or later.

4.4.3 Block operation status check instruction (a, b, &a, &b, la, lb)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J ☐\☐		Intelligent Function Module U☐\G☐	Index Z☐	Constant K, H	Expansion SFC	Other BLm		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓢ	—							—	—	○	Device name	○	—	○	○	—	—



[Function]

(1) Checks whether the specified block is active or inactive.

(2) The contact status changes as described below depending on whether the specified block is active or inactive.

Block Status	Contact of N/O Contact Instruction	Contact of N/C Contact Instruction
Active	ON	OFF
Inactive	OFF	ON

(3) The contact is always OFF if the block that does not exist in the SFC program is specified.

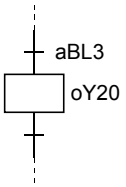
REMARKS

As the “BLm” device is treated as a virtual device, the contact on the monitor of a peripheral device does not turn ON/OFF. If the internal device is ON, the coil instruction is switched ON for operations.

4 - 48

[Program Examples]

(1) The following program turns ON Y20 when block 3 is active.



Related Instructions

a) SFC control instructions
- Block START instruction (sBLm) and block END instruction (rBLm) ..... See Section 4.4.6

b) SFC diagram symbols
- Block START step (Bm ⊟, Bm ⊟) ..... See Sections 4.2.8 and 4.2.9

c) SFC information device
- Block START/END bit ..... See Section 4.5.1

4 - 49

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	△*	○	○	○	○	○

* : The serial number (first five digits) shall be 04122 or later.

4.4.4 Active step batch readout instructions (MOV, DMOV)

	Usable Devices										Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J \		Intelligent Function Module U \G	Index Z	Constant K, H	Expansion SFC BLm\Sn	Other		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓢ	⊛	—						—	○	—	BIN16/ BIN32	○	○	—	—	○	—
Ⓓ	○							—	—	—							

⊛ : Only step relay (S) can be used

[When step in current block is specified]

MOV(P) Ⓢ(K4Sn) Ⓓ

n is a step No.

DMOV(P) Ⓢ(K8Sn) Ⓓ

[When step in another block is specified]
[When step is specified in sequence program]

MOV(P) Ⓢ(BLm\K4Sn) Ⓓ

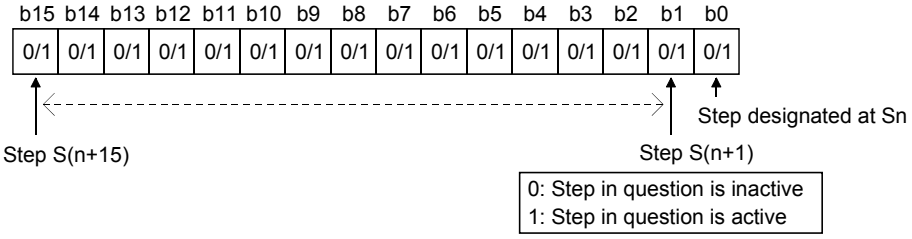
m is a block No. and n is a step No.

DMOV(P) Ⓢ(BLm\K8Sn) Ⓓ

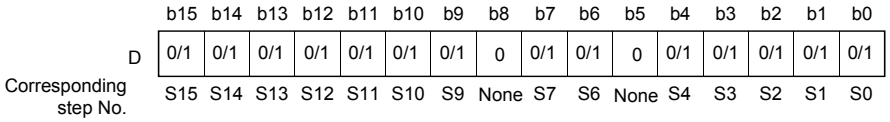
[Function]

(1) Executes a batch readout of the operation statuses (active/inactive) of steps in a specified block.

(2) The readout results are stored at the "Ⓓ" device as shown below.



(3) The bit corresponding to the unassigned step No. (nonexistent step No.) in the read data turns to "0".
When step 5 and step 8 do not exist in the read block, b5 and b8 turn to "0".

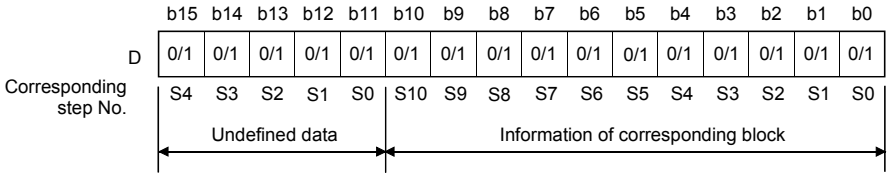


4 - 50

(4) When the block is not specified, specify the step number with which the read data range does not exceed the maximum step No. in the block.

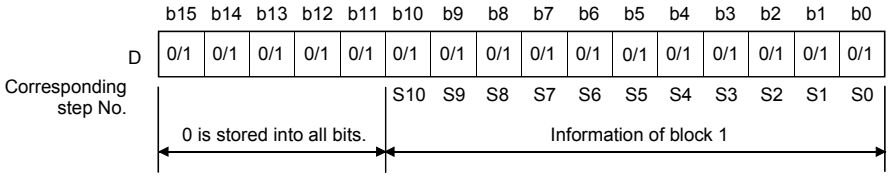
(a) If the maximum number of steps is exceeded, data will be undefined.

For example, when the last step of the block to be read is step 10 (S10), data in b11 to 15 will be undefined.



(b) When the block has been specified, "0" is stored into the remaining bits.

When block 1 is specified, "0" is stored into B11 - 15 if the last step of block 1 is step 10 (S10).



(5) In the activation step batch read instruction, do not specify a nonexistent block/step.

An error will not occur if a nonexistent block/step is specified.

However, the read data are undefined.

The OPERATION ERROR (error code: 4101) will occur in the Universal model QCPU and LCPU if a nonexistent step is specified when the block specification is not performed.

(6) Specify the step as described below.

(a) In the case of SFC program

1) Use "K4Sn" when specifying the step in the current block.

2) Use "BLm\K4Sn" when specifying the step in the SFC program.

(b) In the case of sequence program

1) Use "BLm\K4Sn" when executing the step activation check instruction.

2) When the block number is not specified, specify the block number with the BRSET instruction.

Note, however, that the following CPU modules cannot use the BRSET instruction.

When no block number is specified, the block 0 is set.

• Basic model QCPU

• Universal model QCPU whose serial number (first five digits) is "13101" or earlier

• LCPU

[Operation Error]

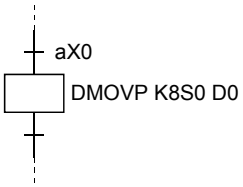
• If exceeding the maximum step No. (8191) when block specification is not made (for the Universal model QCPU or LCPU).....Error No.4101

• If specifying the stop which does not exist when block specification is not made (for the Universal model QCPU or LCPU).....Error No.4101

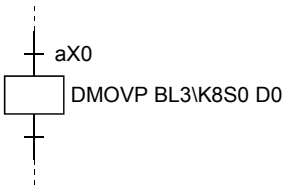
[Program Examples]

(1) The following program reads 32 active steps, starting from step 0 of block 3, to D0 and D1 when X0 turns ON.

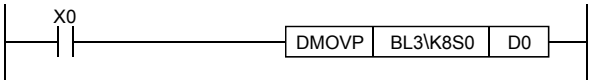
When step is designated by operation output of block 3

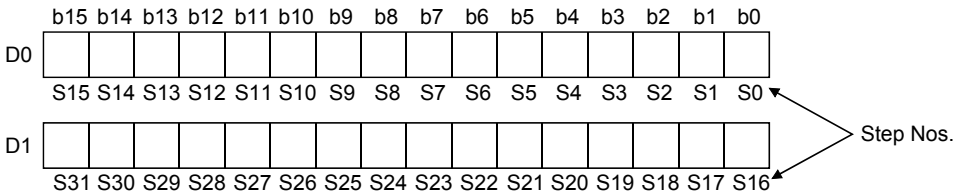


When step is designated by operation output of other than block 3



When step is designated by sequence program





Related Instructions

1) SFC control instructions
- Block switching instruction (BRSET).....See Section 4.4.11.
- Step operation status check instruction (a, b, &a, &b, la, lb).....See Section 4.4.1.
- Active step batch readout instruction (BMOV).....See Section 4.4.5.

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	△*	○	○	○	○	○

* : The serial number (first five digits) shall be 04122 or later.

4.4.5 Active step batch readout (BMOV)

	Usable Devices										Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J ☐\☐		Intelligent Function Module U☐\G☐	Index Z☐	Constant K, H	Expansion SFC BLm\Sn	Other Sn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓢ	⊛	—						—	○	—	BIN16	○	○	—	—	○	—
Ⓓ	○						—	—	—	—							
ⓝ	—							○	—	—							

⊛ : Only step relay (S) can be used

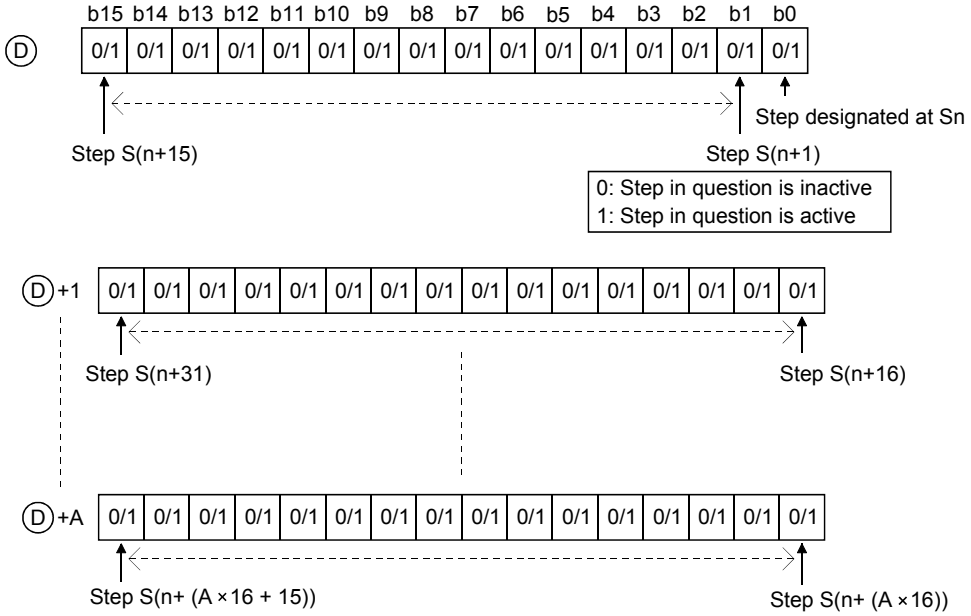
BMOV(P) Ⓢ(K4Sn) Ⓓ ⓝ
└ n is a step No.

BMOV(P) Ⓢ(BLm\K4Sn) Ⓓ ⓝ
└ m is a block No. and n is a step No.

[Function]

(1) A batch readout (designated number of words) of step operation statuses is executed at the specified block.

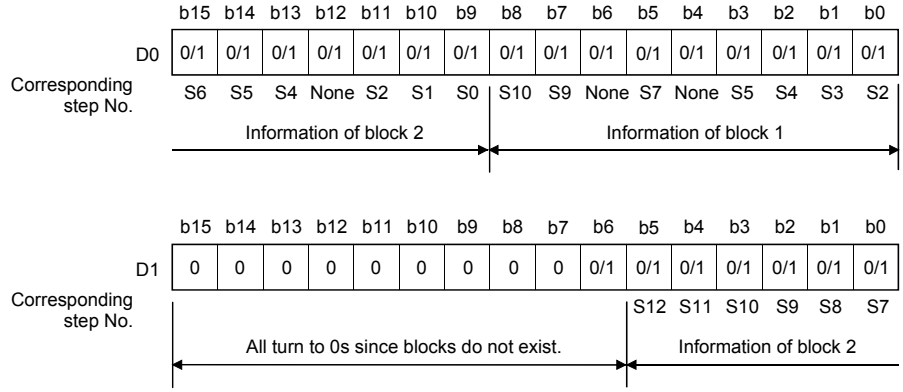
(2) The readout results are stored at the “D” device as shown below.



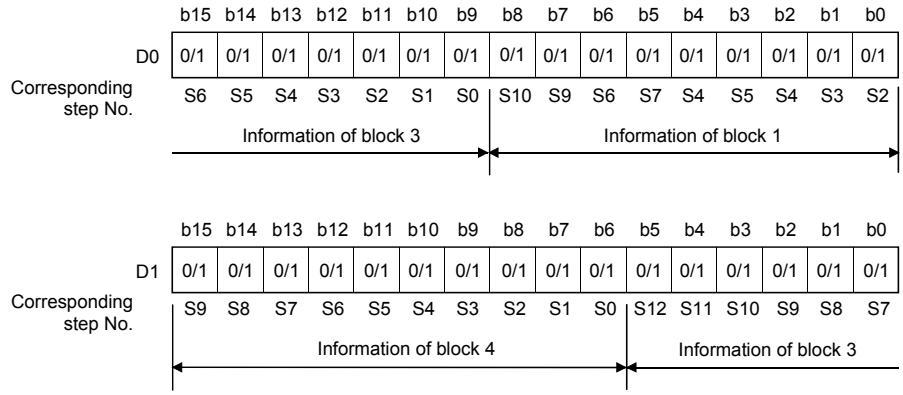
(3) The bit corresponding to the unassigned step No. (nonexistent step No.) in the read data turns to "0".

4 - 53

(4) If the read data range exceeds the maximum step No. in the block, the data of the next block No. are read.
When there are no blocks in and after the block to be read, "0" is stored into the remaining bits.
Example:
When "BMOV BL1\S2 D0 K2" is executed in the following case,
• Block 1 : The maximum step No. is 10 (S10) and step 5 (S5) and step 8 (S8) do not exist
• Block 2 : The maximum step No. is 12 (S12) and step 3 (S3) does not exist
• Block 3 and later: Do not exist
data are stored as shown below.



(5) If there is a nonexistent block in the data to be read, the nonexistent block is omitted and the data of the next existing block are read.
Example:
When "BMOV BL1\S2 D0 K2" is executed in the following case,
• Block 1: The maximum step No. is 10 (S10)
• Block 2: Nonexistent
• Block 3: The maximum step No. is 12 (S12)
• Block 4: The maximum step No. is 15 (S15)
data are stored as shown below.



(6) In the activation step batch read instruction, do not specify a nonexistent block/step.
An error will not occur if a nonexistent block/step is specified.
However, the read data are undefined.

(7) Specify the step as described below.

(a) In the case of SFC program

1) Use "K4Sn" when specifying the step in the current block.

2) Use "BLm\K4Sn" when specifying the step in the SFC program.

(b) In the case of sequence program

1) Use "BLm\K4Sn" when executing the step activation check instruction.

2) When the block number is not specified, specify the block number with the BRSET instruction.

Note, however, that the following CPU modules cannot use the BRSET instruction. When no block number is specified, the block 0 is set.

• Basic model QCPU

• Universal model QCPU whose serial number (first five digits) is "13101" or earlier

• LCPU

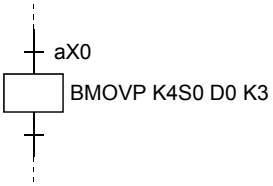
[Operation Error]

• When the step relay (S) range is exceeded .....Error No. 4101

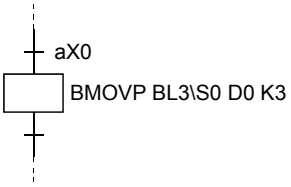
[Program Examples]

(1) The following program reads the active step status of 48 steps (3 words), starting from step 0 of block 3, to D0 - D2 when X0 turns ON.

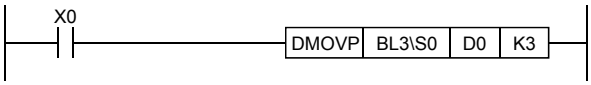
When step is designated by operation output of block 3

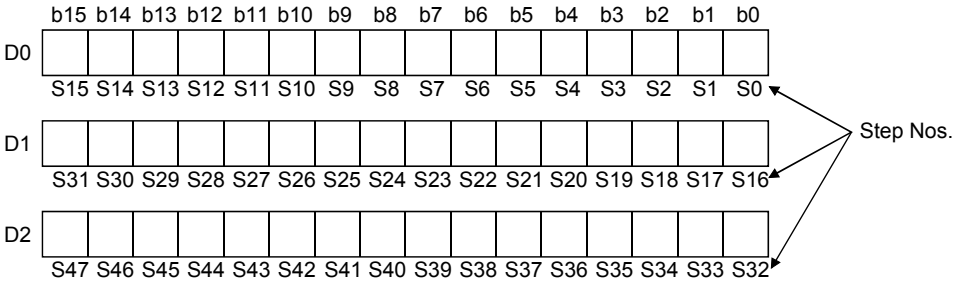


When step is designated by operation output of other than block 3



When step is designated by sequence program





Related Instructions

1) SFC control instructions
   - Block switching instruction (BRSET).....See Section 4.4.11
   - Step operation status check instruction (a, b, &a, &b, la, lb).....See Section 4.4.1
   - Active step batch readout instruction (MOV, DMOV).....See Section 4.4.4

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	△*	○	○	○	○	○

* : The serial number (first five digits) shall be 04122 or later.

4.4.6 Block START & END instructions (s, r)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J □\□		Intelligent Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn BLm\TRn	Other BLm		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓓ	—							—	—	○	Device name	○	○	—	○	—	—

[Function]

(1) Block START instruction (sBLm)

(a) A specified block is forcibly activated independently and is executed from its initial step.
When there are multiple initial steps, all initial steps become active.
When the bock START/END bit of the SFC information devices has been set, the corresponding bit device changes from OFF to ON.

(b) If the specified block is already active when this instruction is executed, the instruction will be ignored (equivalent to the NOP instruction), and processing will continue.

(2) Block END instruction (rBLm)

(a) A specified block is forcibly deactivated independently.
When there are active steps, all are deactivated and the coil outputs are turned OFF.
When the bock START/END bit of the SFC information devices has been set, the corresponding bit device changes from ON to OFF.

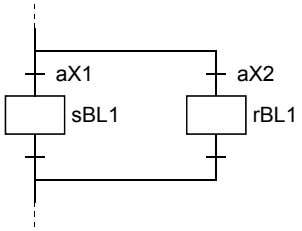
(b) If the specified block is already inactive when this instruction is executed, the instruction will be ignored (equivalent to the NOP instruction) and processing will continue.

[Operation Error]

• Error No. 4621 occurs when the specified block does not exist or when the SFC program is in the stand-by status.

[Program Examples]

(1) When X1 switches ON, the following program forcibly activates block1. When X2 switches ON, it ends and forcibly deactivates block1.



Related Instructions

a) SFC diagram symbols
• Block START step (Bm□, Bm□) .....See Sections 4.2.8 and 4.2.9.

b) SFC information device
• Block START/END bit.....See Section 4.5.1.

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	△*	○	○	○	○	○

* : The serial number (first five digits) shall be 04122 or later.

4.4.7 Block STOP and RESTART instructions (PAUSE, RSTART)

	Usable Devices										Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J ☐\☐		Intelligent Function Module U☐\G☐	Index Z☐	Constant K, H	Expansion SFC BLm\Sn BLm\TRn	Other BLm		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓓ	—							—	—	○	Device name	○	○	—	○	—	—

PAUSE Ⓓ(BLm)
m is a block No.

PSTART Ⓓ(BLm)
m is a block No.

[Function]

(1) Block STOP instruction (PAUSE)

(a) Executes a temporary stop at the specified block.

(b) As shown below, processing varies, depending on when the stop occurs and on the coil output status setting (designated by OUT instruction).

Setting of Output Mode at Block Stop in PLC Parameter	Operation Output at Block Stop (SM325)	Status of STOP-time Mode Bit	Operation			
			Active step other than held step (including HOLD step whose transition condition is not satisfied)	Held step *		
				Coil HOLD step (SC)	Operation HOLD step (without transition check) (SE)	Operation HOLD step (with transition check) (ST)
• Turns OFF (coil output OFF) • Remains ON (coil output held)	• OFF (coil output OFF)	OFF or no setting (immediate stop)	• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. • The status remains active.	• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. • The status becomes inactive.	• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. • The status remains active.	
		ON (STOP after transition)	• Normal operation is performed until the transition condition is satisfied. • When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block is stopped before execution of the operation output.			
• Remains ON (coil output held)	• ON (coil output held)	OFF or no setting (immediate stop)	• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. • The status remains active.	• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. • The status remains active.		
		ON (STOP after transition)	• Normal operation is performed until the transition condition is satisfied. • When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block is stopped before execution of the operation output.			

*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

4 - 58

POINTS

The operation of SM325 differs depending on the CPU module.

- For the Basic model QCPU, High Performance model QCPU, and Process CPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) at STOP RUN of the CPU module.
- For the Universal model QCPU and LCPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) when the CPU module is powered ON power or is reset.

Parameter setting	SM325
Turns OFF (coil output OFF)	OFF
Remain ON (coil output held)	ON

Note that the output mode at block stop can be changed regardless of the parameter setting by turning ON/OFF SM325 in the user program.

(c) The STOP/RESTART bit switches ON when the SFC control “block STOP” instruction (PAUSE BLm) is executed.

(2) Block RESTART instruction (RSTART)

(a) The block in question is restarted from the step where a STOP occurred.
An “operation HOLD status” step (with transition check or without transition check) which has been stopped will be restarted with the operation HOLD status in effect.
A “coil output HOLD” step cannot be restarted after being stopped as it becomes deactivated at that time.

(b) Depending on the ON/OFF status of the "block STOP-time operation output flag (SM325)", the operations of the PLS instruction and P instruction after block STOP cancellation change.
- When SM325 is ON (coil output held).....Not executed
- When SM325 is OFF (coil output OFF) .....Executed again

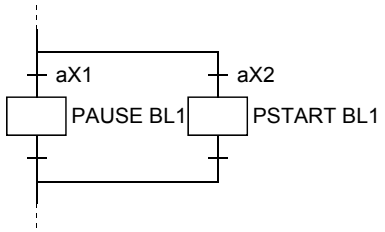
(c) When the block STOP/RESTART bit of the SFC information devices has been set, the block STOP/RESTART bit also turns OFF.

[Operation Error]

- Error No. 4621 occurs when the specified block does not exist or when the SFC program is in the stand-by status.

[Program Examples]

(1) Block 1 is stopped when X1 switches ON, and is restarted when X2 switches ON.



4 - 59

Related Instructions

1) SFC information device
   • Block STOP/RESTART bit .....See Section 4.5.3.

4 - 60

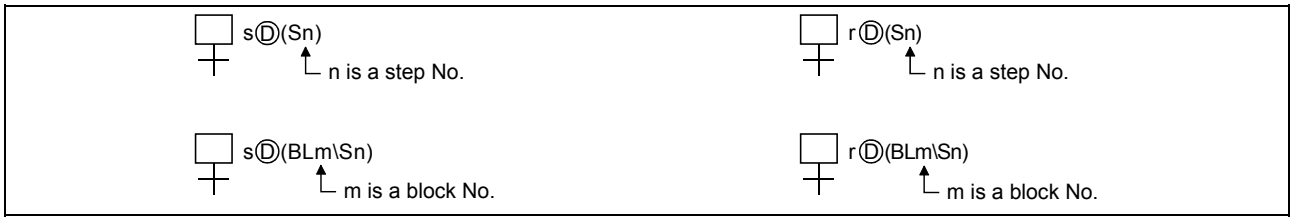
Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	△*	○	○	○	○	○

* : The serial number (first five digits) shall be 04122 or later.

4.4.8 Step START and END instructions (s, r)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J □\□		Intelligent Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn	Other Sn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓓ	⊛	—						—	○	—	Device name	○	○	—	—	○	—

⊛ : Only step relay (S) can be used



[Function]

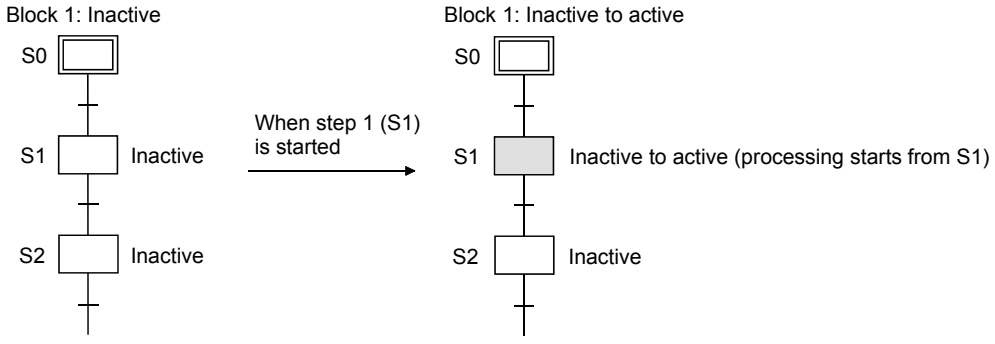
(1) Step START instruction (s)

(a) A specified step at a specified block is activated forcibly. Operation at the block in question varies as follows, depending on whether the block is active or inactive.

1) When the specified block is inactive:

The specified block is activated when the instruction is executed, and processing starts from the specified step.

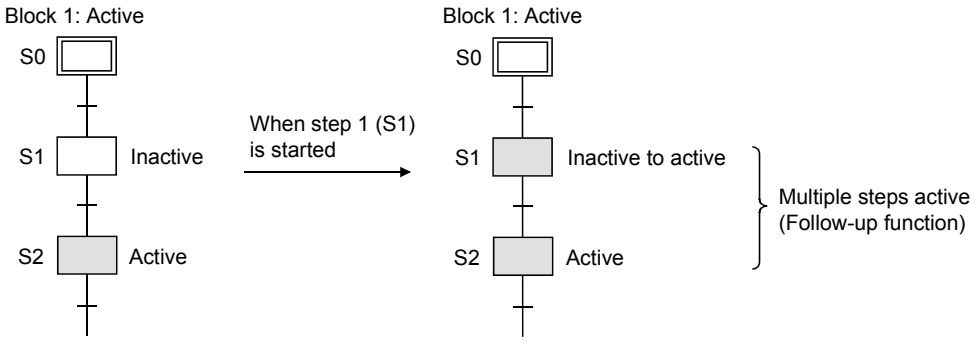
Processing is performed as shown below when step 1 in block 1 is started in the sequence program.



When the block START/END bit of the SFC information devices has been set, the corresponding bit device changes from OFF to ON.

4 - 61

2) When the specified block is active:
If the step is already active when the SET instruction is executed, the step will remain active and processing will continue, with another step being designated as active. (Multiple step activation, follow-up function.)
Processing is performed as shown below when step 1 in block 1 is started in the sequence program.

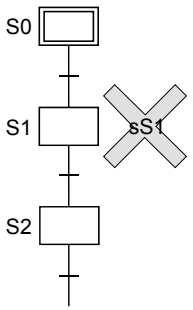


(b) When multiple initial steps exist, an initial step selection START will occur when a given step is specified and activated.

(c) When designating a step located in a parallel branch, all the parallel steps should be activated. An inactive parallel branch ladder at such a time will prevent the parallel coupling condition from being satisfied.

(d) If a specified step is already active when this instruction is executed, the instruction will be ignored (equivalent to the NOP instruction), and processing will continue.
To hold a specified step with the HOLD step, see "Transition to HOLD step by double START" in Section 4.7.6.

(e) When the operation output is used to start the step, do not specify the current step number as the specified step number.
If the current step is designated as the specified step number, normal operation will not be performed.



(f) Specify the step as described below.
1) In the case of SFC program
• Use "Sn" when specifying the step in the current block.
• Use "BLm\Sn" when specifying the step in another block.
2) In the case of sequence program
• Use "BLm\Sn" when executing the step START instruction in the sequence program.
• When the block number is not specified, specify the block number with the BRSET instruction.
Note, however, that the following CPU modules cannot use the BRSET instruction.
When no block number is specified, the block 0 is set.
• Basic model QCPU
• Universal model QCPU whose serial number (first five digits) is "13101" or earlier
• LCPU

(2) Step END instruction (r)

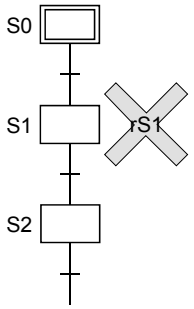
(a) A specified step at a specified block is forcibly deactivated. “Coil HOLD” and “operation HOLD” steps are subject to this instruction.

(b) When the number of active steps in the corresponding block reaches 0 due to the execution of this instruction, END step processing is performed and the block becomes inactive.
When the bock START/END bit of the SFC information devices has been set, the corresponding bit device changes from ON to OFF.

(c) If the RST instruction is executed at a step located in a parallel branch, the parallel coupling condition will remain unsatisfied.

(d) If a specified step is already inactive when this instruction is executed, the instruction will be ignored (equivalent to the NOP instruction).

(e) When the operation output is used to end the step, do not specify the current step as the specified step number.
If the current step is designated as the specified step number, normal operation will not be performed.



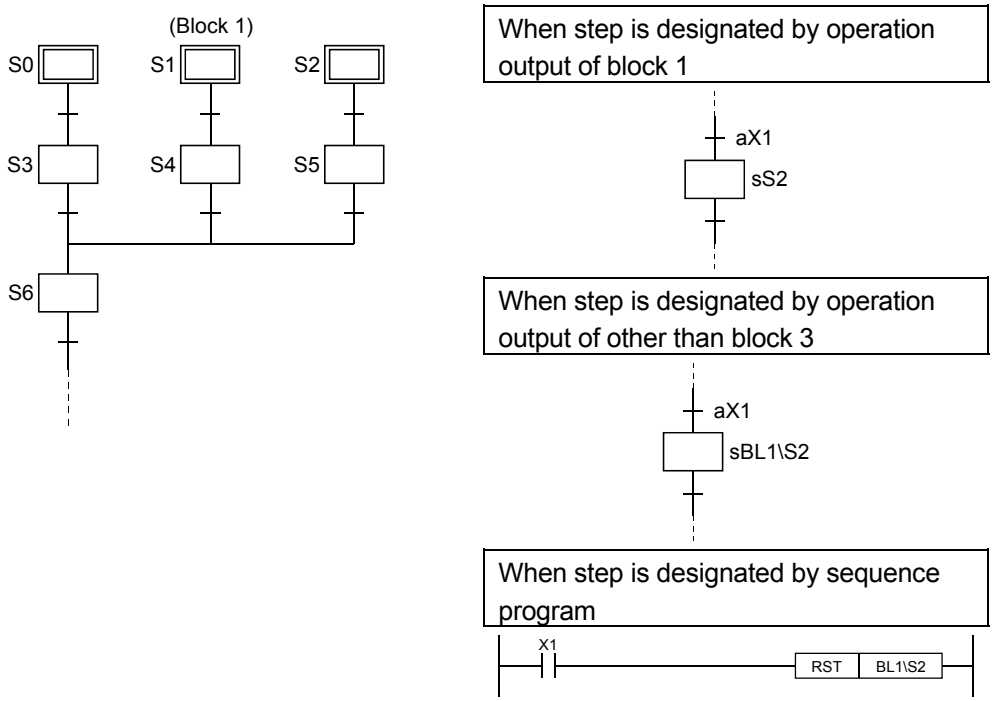
(f) Specify the step as described below.
1) In the case of SFC program
• Use "Sn" when specifying the step in the current block.
• Use "BLm\Sn" when specifying the step in another block.
2) In the case of sequence program
• Use "BLm\Sn" when executing the step END instruction in the sequence program.
• When the block number is not specified, specify the block number with the BRSET instruction.
Note, however, that the following CPU modules cannot use the BRSET instruction.
When no block number is specified, the block 0 is set.
• Basic model QCPU
• Universal model QCPU whose serial number (first five digits) is "13101" or earlier
• LCPU

[Operation Error]

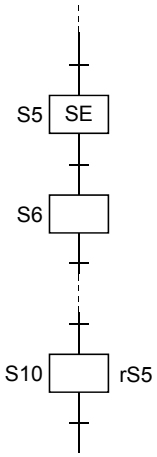
• When no specified step is present or the SFC program is in stand-by mode: Error No.4631
• If using the own step as the specification step No. (Basic model QCPU, Universal model QCPU, and LCPU) ..... Error No.4505

[Program Examples]

(1) When X1 switches ON, the following program will select and start step 2 of block 1 which contains multiple initial steps.



(2) The following program deactivates held step 5 when step 10 is activated.

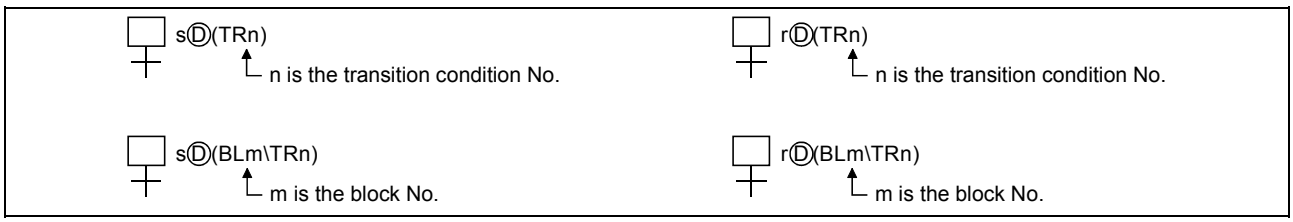


4 - 64

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	×	○	×	○	○	×

4.4.9 Forced transition EXECUTE & CANCEL instructions (s, r)

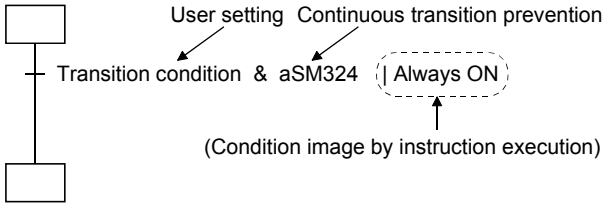
	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J \		Intelligent Function Module U \G	Index Z	Constant K, H	Expansion SFC BLm\TRn	Other TRn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓓ	—							—	○	○	Device name	○	○	—	—	—	○



[Function]

(1) Forced transition EXECUTE instruction (s)

(a) A specified transition condition in a specified block is forcibly satisfied, and an unconditional transition is executed at the step which precedes the condition.



(b) After execution of the instruction, the forced transition status remains effective until a reset instruction is executed.

(2) Forced transition CANCEL instruction (r)

(a) Cancels the forced transition setting (designated by SET instruction) at a transition condition, and restores the transition condition ladder created by the user.

(3) Specify the transition condition as described below.

(a) In the case of SFC program

- Use "TRn" when specifying the transition condition in the current block.
- Use "BLm \TRn" when specifying the transition condition in another block.

(b) In the case of sequence program

- Use "BLm \TRn" when executing the forced transition EXECUTE/CANCEL instruction in the sequence program.
- When the block number is not specified, specify the block number with the BRSET instruction.

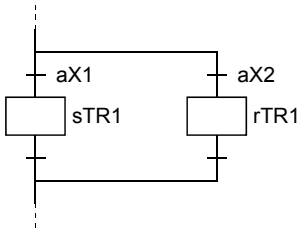
[Operation Error]

• When the specified transition condition does not exist or the SFC program is in a wait state ..... Error No. 4631

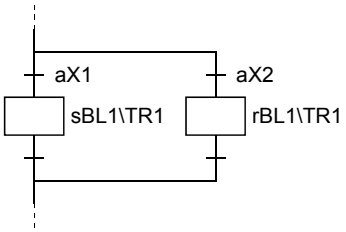
[Program Examples]

(1) When X1 switches ON, the following program executes a forced transition at transition condition 1 of block 1. The forced transition setting is canceled when X2 switches ON.

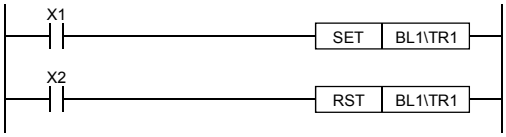
When step is designated by operation output of block 1



When step is designated by operation output of other than block 1



When step is designated by sequence program



POINTS

This instruction checks, from the first sequence step of the specified block in series, whether or not the specified transition condition number is existed.
Because of this, processing time of the instruction differs depending on the program capacity of the specified block (number of sequence steps), a maximum of hundred and several tens ms may be taken.
In case of occurring WDT error (error code: 5001), change the WDT setting value with the PLC RAS setting in the PLC parameter.

4 - 66

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	×	○	×	○	○	×

4.4.10 Active step change instruction (SCHG)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J □\		Intelligent Function Module U□\G□	Index Z□	Constant	Expansion SFC	Other		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓓ	○							○	—	—	BIN16	—	○	—	—	○	—

SCHGⒹ

[Function]

(1) Deactivates the step that executed an instruction, and forcibly activates the specified step (set with the device designated by Ⓓ) in the same block.

(2) When the destination step is already active, the step that executed the SCHG instruction is deactivated and the destination step continues processing as-is.

(3) The step where this instruction is executed is deactivated when processing proceeds to the transition condition status check following the completion of that step's program operation.

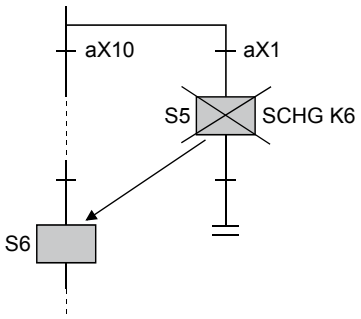
(4) This instruction can only be used at SFC program steps.

[Operation Error]

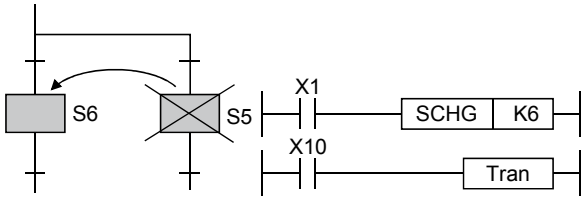
- Error No.4631 occurs when the specified destination step does not exist.
- Error No.4001 occurs when this instruction is used at a sequence program other than an SFC program (error is activated on switching from STOP to RUN).

[Program Examples]

(1) The following program causes a transition as-is when X10 has turned ON before X1 turns ON, and deactivates step 5 and activates step 6 when X1 has turned ON before X10 turns ON.



The program created with MELSAP3 is as shown below.



4 - 67

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	×	○	△*	○	○	×

* : The serial number (first five digits) shall be 13102 or later.

4.4.11 Block switching instruction (BRSET)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J ☐\☐		Intelligent Function Module U☐\G☐	Index Z☐	Constant K, H	Expansion SFC	Other		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓢ	○							○	—	—	BIN16	○	○	—	—	—	—

BRSET Ⓢ

[Function]

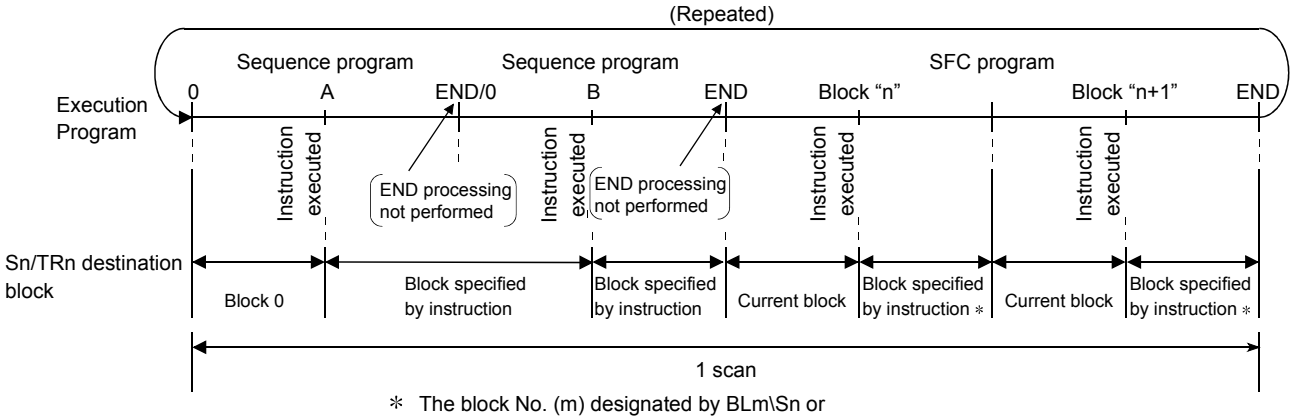
(1) Switches the target block number of the SFC control instruction that specifies only a step (Sn) and transition condition (TRn) to the number set for the device designated by Ⓢ.

(2) Although “BLm\Sn” or “BLm\TRn” may be used as the instruction device when designating the destination block number, only a constant (K, H) may be designated at the “m” of “BLm”, thereby fixing the designation destination.
When block switching is executed by this BRSET instruction, a word device can be used for indirect designation, index modification, etc.

(3) The effective operation range when block switching occurs (by BRSET instruction) varies according to the program being run at the time, as shown below.
1) When this instruction is executed in a sequence program, target block switching is valid from instruction execution to SFC execution.
At the next scan, the target block is block 0 as the default until the instruction is executed again.

4 - 68

2) If the BRSET instruction is executed at an SFC program, block switching will be effective only for the step currently being executed.
Even if the step in question is the same step, the BRSET instruction must be executed at each block where the Sn and TRn instructions are used.
Moreover, within a single step, block switching will be effective from the point where the BRSET instruction is executed to that step's processing END point.
When processing is repeated at the next scan following the processing END for that step, the block in question will be designated as the “current block” until the point when the BRSET instruction is executed again.



∗ The block No. (m) designated by BLm\Sn or BLm\TRn becomes valid regardless of whether this instruction is executed or not.

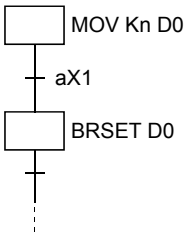
∗ When multiple steps are active at parallel branch, etc., only the step where the instruction was executed will be valid.
When it is desired to designate blocks at multiple steps, the instruction must be executed at each step.

[Operation Error]

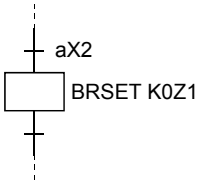
• Error No. 4621 occurs when the specified block does not exist or when the SFC program is in the stand-by status.

[Program Examples]

(1) When X1 switches ON, the following program switches the Sn or TRn block number to the block number stored at the D0 data register.



(2) When X2 switches ON, the following program switches the Sn or TRn block number according to the constant at the Z1 index register.



4 - 69

4.5 SFC Information Devices

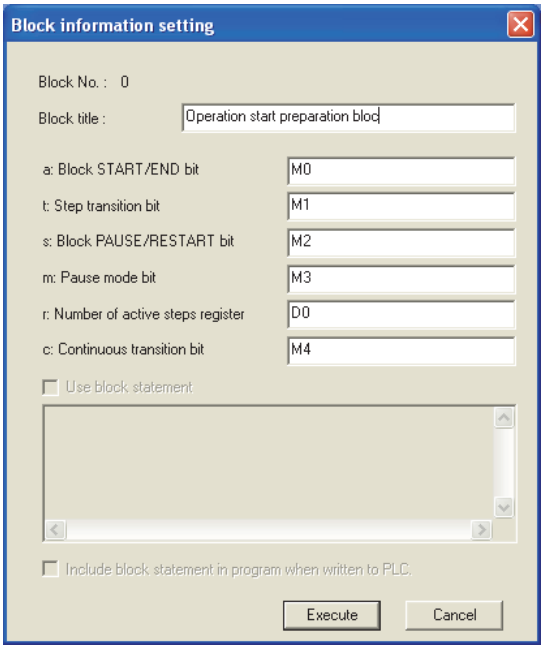
This section explains the SFC information devices set in each block. Table 4.2 indicates the SFC information device types and usable devices.

Table 4.3 SFC Information Device List

SFC Information Device	Function Outline	Usable Device	QCPU, LCPU
Block START/END bit	• Device designed to forcibly start or forcibly end the specified block by a sequence program or the test operation of the peripheral device. • Can also be used to confirm the active status of the specified block.	Y, M, L, F, V, B	○
Step transition bit	• Device that checks whether or not a step transition occurred in the corresponding scan in the specified block.		
Block STOP/ RESTART bit	• Device designed to stop temporarily or restart the corresponding block that is active.		
Block STOP mode bit	• Device used to specify whether all steps will be immediately stopped or the block will be stopped after the transition of the corresponding step when the block is stopped temporarily.		
Continuous transition bit	• Device used to specify whether the operation output of the next step will be executed within the same scan or not when the transition condition is satisfied.		
"Number of active steps" register	• Device that stores the number of steps currently active in the specified block.	D, W, R, ZR	○

○: Usable

When using the SFC information devices, set them in "Block information setting" at the input (editing) of the SFC diagram. When the SFC information devices are not used, they need not be set.



4 - 70

POINTS

The following cannot be specified for the SFC information devices.

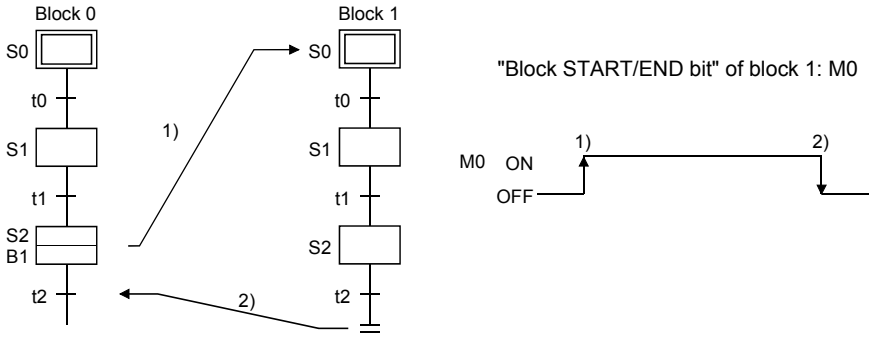
- Indirect designation (@)
- Digit designation (K)
- Index modification (Z)
- Word device bit designation (.)

4 - 71

4.5.1 Block START/END bit

The block START/END bit is used to confirm the active status of the specified block by a sequence program or the test operation of the peripheral device.
It can also be used as a device to forcibly start or forcibly end the specified block.

(1) Operation of block START/END bit
  (a) The block START/END bit turns ON when the corresponding block starts.
    The block START/END bit remains ON while the corresponding block is active.
  (b) The block START/END bit turns OFF when the corresponding block becomes inactive.
    The block START/END bit remains OFF while the corresponding block is inactive.



(2) When the corresponding block is inactive, it can be started independently by forcibly turning ON the block START/END bit.
  While the corresponding block is active, the processing of the corresponding block can be forcibly ended by forcibly turning OFF the block START/END bit.
  The block START/END bit can also be turned ON/OFF in the test mode of the peripheral device.

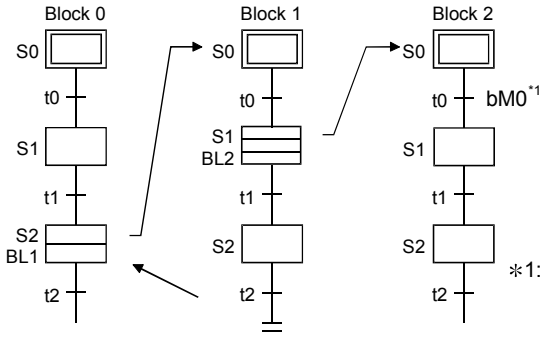
(3) When a forced OFF is executed by the block START/END bit, and the block in question becomes inactive, processing will occur as follows:
  (a) Execution of the block in question will stop together with all outputs from the step which was being executed. (Devices switched ON by the SET instruction will not switch OFF.)
  (b) If another block is being started by the block START step in the corresponding block, the corresponding block stops.
    However, the start destination block remains active and continues processing.
    To also end the start destination block simultaneously, the block START/END bit of the start destination must also be turned OFF.

(4) A block which has been forcibly deactivated is restarted as shown below.

Relevant Block		Restart Status
Block 0	When the START condition of block 0 is "Auto START ON" in the SFC setting of the PLC parameter dialog box.	Operation is restarted from the initial step following END step processing.
	When the START condition of block 0 is "Auto START OFF" in the SFC setting of the PLC parameter dialog box.	The block is deactivated after END step processing, and processing is restarted from the initial step when another START request occurs for that block.
Other than block 0		

[Program Example]

Use the contact of the "block START/END bit" when a transition occurs after block 1 ends.



*1: Using "BL1" instead of "block START/END bit" as contact, the same operation can be performed.

Related Instructions

1) SFC control instructions
   • Block START instruction (sBLm), block END instruction (rBLm).....See Section 4.4.6.
2) SFC diagram symbols
   • Block START step (Bm□, Bm▤).....See Sections 4.2.8.and 4.2.9.

4 - 73

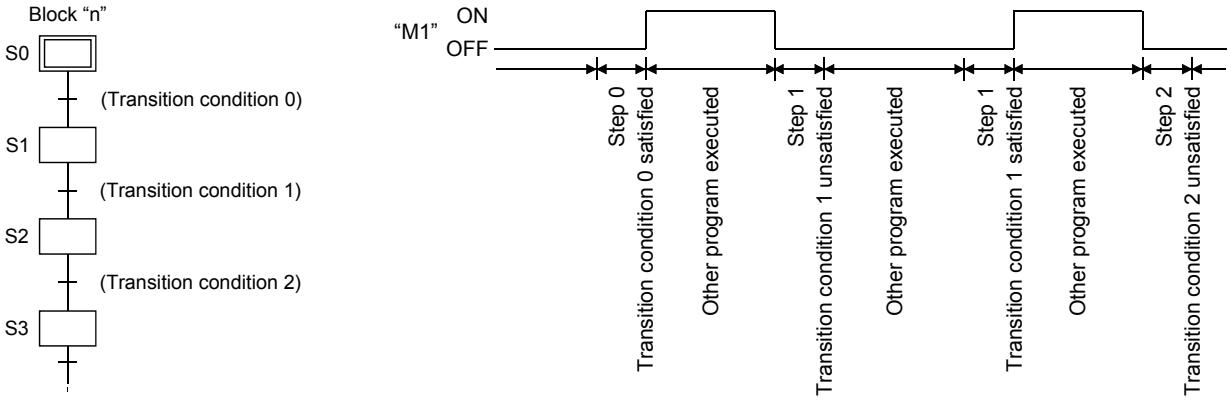
4.5.2 Step transition bit

The step transition bit is designed to check whether the transition condition of the step in execution has been satisfied or not.

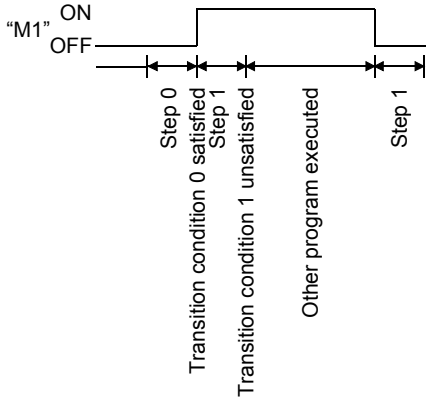
(1) After the operation output at each step is completed, the step transition bit automatically switches ON when the transition condition (for transition to the next step) is satisfied.

(2) A transition bit which is ON will automatically switch OFF when processing of the block in question occurs again.

Example: Step transition bit = M1

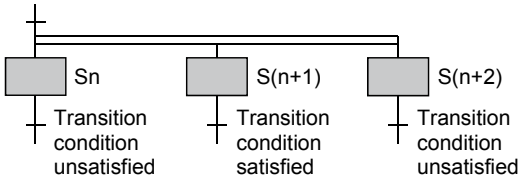


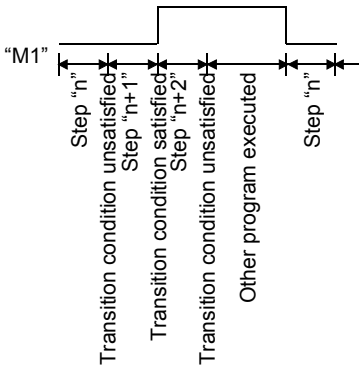
(3) If a continuous transition is designated (continuous transition bit ON), the transition bit will remain ON during the next step's operation output after the transition condition is satisfied.
It will also remain ON following the execution of multiple steps, even if the transition condition is unsatisfied.
In these cases, the transition bit will switch OFF when block execution occurs at the next scan.
Example: Step transition bit = M1



4 - 74

(4) At active parallel branch steps, the transition bit will switch ON when any of the transition conditions are satisfied.





4 - 75

4.5.3 Block STOP/RESTART bit

The block STOP/RESTART bit is used to temporarily stop processing while the corresponding block is active.

(1) When the designated block STOP/RESTART bit is switched ON by the sequence program or peripheral device, processing will be stopped at the current step of the block in question.
If a START status is in effect at another block, the STOP will still occur, but the START destination block will remain active and processing will continue.
To stop the START destination block at the same time, the START destination's block STOP/RESTART bit must also be switched OFF.

(2) When a block is stopped by switching the block STOP/RESTART bit ON, the STOP timing will be as shown below.

<table>
<tr><th rowspan="3">Setting of Output Mode at Block Stop in PLC Parameter</th><th rowspan="3">Operation Output at Block Stop (SM325)</th><th rowspan="3">Status of STOP-time Mode Bit</th><th colspan="4">Operation</th></tr>
<tr><th rowspan="2">Active step other than held step (including HOLD step whose transition condition is not satisfied)</th><th colspan="3">Held step *</th></tr>
<tr><th>Coil HOLD step (SC)</th><th>Operation HOLD step (without transition check) (SE)</th><th>Operation HOLD step (with transition check) (ST)</th></tr>
<tr><td rowspan="2">• Turns OFF (coil output OFF)
• Remains ON (coil output held)</td><td rowspan="2">• OFF (coil output OFF)</td><td>OFF or no setting (immediate stop)</td><td>• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status remains active.</td><td rowspan="2">• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status becomes inactive.</td><td colspan="2" rowspan="2">• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status remains active.</td></tr>
<tr><td>ON (STOP after transition)</td><td>• Normal operation is performed until the transition condition is satisfied.
• When the transition condition is satisfied, the end processing of the corresponding step is performed.
At the same time, the transition destination step becomes active and the block stops immediately.</td></tr>
<tr><td rowspan="2">• Remains ON (coil output held)</td><td rowspan="2">• ON (coil output held)</td><td>OFF or no setting (immediate stop)</td><td>• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held.
• The status remains active.</td><td colspan="3" rowspan="2">• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held.
• The status remains active.</td></tr>
<tr><td>ON (STOP after transition)</td><td>• Normal operation is performed until the transition condition is satisfied.
• When the transition condition is satisfied, the end processing of the corresponding step is performed.
At the same time, the transition destination step becomes active and the block stops immediately.</td></tr>
</table>

*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

4 - 76

POINTS

The operation of SM325 differs depending on the CPU module.

- For the Basic model QCPU, High Performance model QCPU, and Process CPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) at STOP RUN of the CPU module.
- For the Universal model QCPU and LCPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) when the CPU module is powered ON or is reset.

Parameter setting	SM325
Turns OFF (coil output OFF)	OFF
Remain ON (coil output held)	ON

Note that the output mode at block stop can be changed regardless of the parameter setting by turning ON/OFF SM325 in the user program.

4 - 77

(3) The execution of the corresponding block is restarted from the step where it had stopped when the "block STOP/RESTART bit" is turned OFF in the sequence program, SFC program or peripheral device.
An “operation HOLD status” step (with transition check or without transition check) which has been stopped will be restarted with the operation HOLD status in effect.
A coil output HOLD step cannot be restarted after being stopped as it is deactivated at that time.

(4) When a block STOP is canceled, the PLS or P instruction is executed.
When the special relay for operation output selection at block STOP (SM325) is turned ON, the PLS or P instruction is not executed if a block STOP is canceled.

(5) When the SFC control “block STOP” instruction (PAUSE BLm) is executed, the block in question is stopped, and the block STOP/RESTART bit switches ON.
When the “block RESTART” instruction (RSTART BLm) is executed while the block is stopped, the block in question is restarted, and the block STOP/RESTART bit switches OFF.

POINTS

(1) Stopping of program processing by a block STOP/RESTART bit being switched ON, or by a block STOP instruction, applies only to the specified block.

(2) Even if a block stop is executed for the START destination block, the START source block will not be stopped.

(3) Even if a block stop is executed for the START source block, the START destination block will not be stopped.

Related Instructions

1) SFC information device
• Block STOP mode bit.....See Section 4.5.4.

2) SFC control instructions
• Block STOP instruction (PAUSE BLm) and block RESTART instruction (RSTART BLm).....See Section 4.4.7.

4 - 78

4.5.4 Block STOP mode bit

The block STOP mode bit setting determines when the specified block is stopped after the block STOP/RESTART bit switches ON, or after a stop designation by the block STOP instruction (PAUSE BLm).

(1) The stop timing for a block where a STOP request has occurred varies according to the ON/OFF setting of the block STOP mode bit, as shown below.

OFF	• The block is stopped immediately when the block STOP/RESTART bit switches from OFF to ON, or when a block STOP instruction is executed. However, if the block STOP/RESTART bit is switched ON within the current block, the STOP will occur when that block is processed at the next scan, or when the instruction is executed.
ON	• The block is stopped at the step transition which occurs when the transition condition for the current step (active step) is satisfied. However, the operation output will not be executed for the step following the transition. • When multiple steps are active in a parallel branch, the STOP will occur sequentially at each of the steps as their transition conditions are satisfied. However, the held step stops immediately after a STOP request independently of the block STOP mode.

(2) When the corresponding block is stopped, the stop timing is as described below.

<table>
<tr><th rowspan="3">Setting of Output Mode at Block Stop in PLC Parameter</th><th rowspan="3">Operation Output at Block Stop (SM325)</th><th rowspan="3">Status of STOP-time Mode Bit</th><th colspan="4">Operation</th></tr>
<tr><th rowspan="2">Active step other than held step (including HOLD step whose transition condition is not satisfied)</th><th colspan="3">Held step *</th></tr>
<tr><th>Coil HOLD step (SC)</th><th>Operation HOLD step (without transition check) (SE)</th><th>Operation HOLD step (with transition check) (ST)</th></tr>
<tr><td rowspan="2">• Turns OFF (coil output OFF)
• Remains ON (coil output held)</td><td rowspan="2">• OFF (coil output OFF)</td><td>OFF or no setting (immediate stop)</td><td>• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status remains active.</td><td rowspan="2">• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status becomes inactive.</td><td colspan="2" rowspan="2">• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status remains active.</td></tr>
<tr><td>ON (STOP after transition)</td><td>• Normal operation is performed until the transition condition is satisfied.
• When the transition condition is satisfied, the end processing of the corresponding step is performed.
At the same time, the transition destination step becomes active and the block is stopped before execution of the operation output.</td></tr>
<tr><td rowspan="2">• Remains ON (coil output held)</td><td rowspan="2">• ON (coil output held)</td><td>OFF or no setting (immediate stop)</td><td>• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held.
• The status remains active.</td><td colspan="3" rowspan="2">• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held.
• The status remains active.</td></tr>
<tr><td>ON (STOP after transition)</td><td>• Normal operation is performed until the transition condition is satisfied.
• When the transition condition is satisfied, the end processing of the corresponding step is performed.
At the same time, the transition destination step becomes active and the block is stopped before execution of the operation output.</td></tr>
</table>

*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

4 - 79

POINTS

The operation of SM325 differs depending on the CPU module.

- For the Basic model QCPU, High Performance model QCPU, and Process CPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) at STOP RUN of the CPU module.
- For the Universal model QCPU and LCPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) when the CPU module is powered ON or is reset.

Parameter setting	SM325
Turns OFF (coil output OFF)	OFF
Remain ON (coil output held)	ON

Note that the output mode at block stop can be changed regardless of the parameter setting by turning ON/OFF SM325 in the user program.

4 - 80

Related Instructions

1) SFC information device
   • Block STOP/RESTART bit .....See Section 4.5.3.

2) SFC control instruction
   • Block STOP instruction (PAUSE BLm) .....See Section 4.4.7.

4 - 81

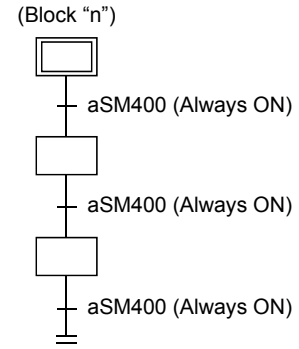
4.5.5 Continuous transition bit

The continuous transition bit specifies whether the operation output of the next step will be executed in the same scan or not when the transition condition is satisfied.

(1) There are two types of SFC program transition processing: "with continuous transition" and "without continuous transition".
The user specifies either of them by turning ON/OFF the continuous transition bit.

- Continuous transition ON (Continuous transition bit: ON)
  ..... When the transition conditions at contiguous steps are satisfied, all the steps transition conditions will be executed at once within a single scan.
- Continuous transition OFF (Continuous transition bit: OFF)
  .....Steps are executed in a 1-step-per-scan format.

Example: Sample program processing



- Continuous transition ON
  When the corresponding block becomes active, the processings of all steps are executed in the same scan, and end step processing is performed to deactivate the block.
- Continuous transition OFF
  When the corresponding block becomes active, steps are executed in a 1-step-per-scan format, and end step processing is performed in the third scan to deactivate the block.

(2) A continuous transition can be designated for individual blocks by the continuous transition bit ON/OFF setting, or for all blocks using the batch setting special relay.
As indicated below, whether a continuous transition is executed or not changes depending on the combination of the continuous transition bit and the special relay that sets "whether continuous transition of all blocks is executed or not" (SM323).

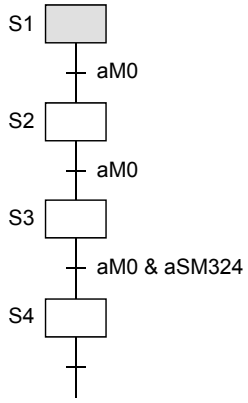
SM323 status	Continuous Transition Bit Status	SFC Program Operation
ON	• Continuous transition bit OFF	• Operation occurs without continuous transition
	• No continuous transition bit setting	• Operation occurs with continuous transition
	• Continuous transition bit ON	
OFF	• Continuous transition bit OFF	• Operation occurs without continuous transition
	• No continuous transition bit setting	
	• Continuous transition bit ON	• Operation occurs with continuous transition

POINT

The tact time can be shortened by setting "with continuous transition".
This resolves the problem of waiting time from when the transition condition is satisfied until the operation output of the transition destination step is executed.
However, when "with continuous transition" is set, the operations of the other blocks and sequence program may become slower.

(3) The continuous transition disable flag (SM324) is always ON (turned ON automatically by the system at SFC program execution) normally, but is OFF during continuous transition.
Use of SM324 under the AND condition in a transition condition disables a continuous transition.

(Example)
[SFC program]

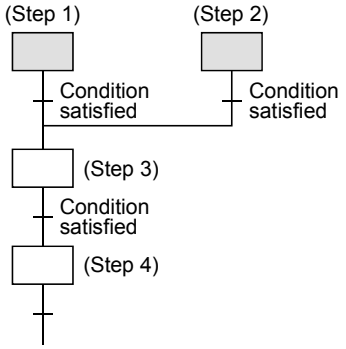


[Operation]

1) When M0 is ON, step 1 to step 4 are the targets of continuous transition.
2) Since SM324 is added as the AND condition to the transition condition following step 3, the transition condition following step 3 is not satisfied after execution of step 3.
3) When step 3 is executed in the next scan, execution proceeds to step 4 in the same scan since SM324 is ON.

POINT

(1) When a jump transition or selection coupling causes a transition from multiple steps to one step, the operation output of one step may be executed twice in a single scan.



When the setting is "with continuous transition" in the case as shown on the left, execution passes through step 3 twice in a single scan.

(2) In the case of "with continuous transition", a step start/end is made within one scan. Since the END processing is not executed in this case, the coil output turned on by the OUT instruction in the operation output is not reflected on the device. When the coil output is the Y output, actual output is not provided. In addition, ON of the step relay cannot be detected.

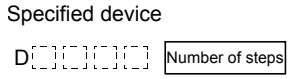
(3) In the case of a program that uses a jump transition for looping, care must be taken when the transition conditions in the loop are all satisfied during execution at the "with continuous transition" setting, since an endless loop will occur within one scan, resulting in WDT Err. (No. 5001).

4 - 83

4.5.6 “Number of active steps” register

The “number of active steps” value for a given block is stored at this register.

(1) The “number of active steps” value for a given block is stored.



(2) The number of active steps applies to the following steps.

- Normal active steps
- Coil HOLD steps
- Operation HOLD steps (without transition check)
- Operation HOLD steps (with transition check)
- Stopping steps
- Step double START waiting steps

4 - 84

4.6 Step Transition Watchdog Timer

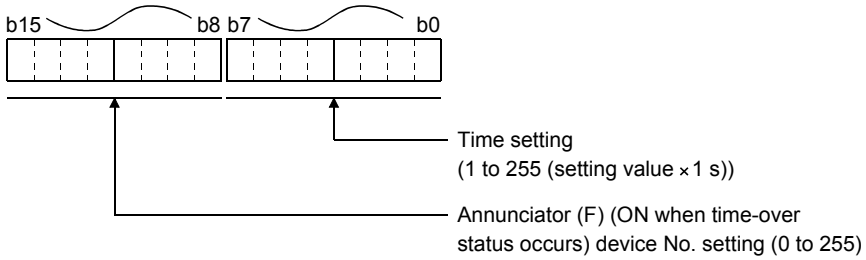
The step transition watchdog timers are timers that measure the time from the point when the relevant step is placed in the execution status until the point when a transition to the next step occurs.
If a transition from the relevant step to the next step fails to occur within the designated time period, the preset annunciator (F) will be turned ON.

(1) When using the step transition watchdog timer, set the "set time" and the "device number of annunciator (F) that will turn ON at time-out" to the special register for step transition watchdog timer setting (SD90 to SD99).
The step transition watchdog timer starts timing when the special relay for step transition watchdog timer start (SM90 to SM99) is turned ON in the operation output of the step that performs a time check.
When any corresponding one of SM90 to SM99 is turned OFF during timing, the step transition watchdog timer stops timing and is reset.

(2) There are 10 step transition watchdog timers, watchdog timer 1 to watchdog timer 10, in the whole SFC program.
The special relay for step transition watchdog timer start and the special register for step transition watchdog timer setting are assigned to each watchdog timer as indicated below.

	Watchdog Timer 1	Watchdog Timer 2	Watchdog Timer 3	Watchdog Timer 4	Watchdog Timer 5	Watchdog Timer 6	Watchdog Timer 7	Watchdog Timer 8	Watchdog Timer 9	Watchdog Timer 10
Special relay	SM90	SM91	SM92	SM93	SM94	SM95	SM96	SM97	SM98	SM99
Special register	SD90	SD91	SD92	SD93	SD94	SD95	SD96	SD97	SD98	SD99

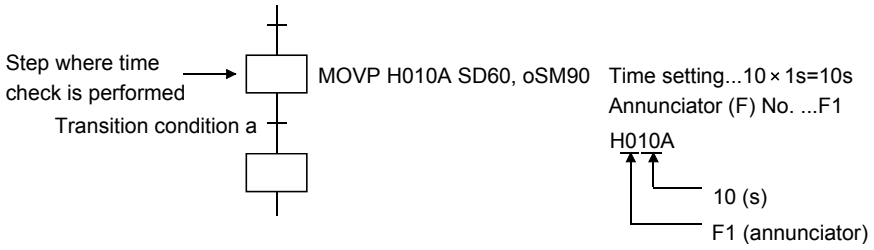
(3) The method of setting to SD90 - SD99 is as shown below.



POINT

(1) When the parameter where the "High speed interrupt I49 fixed scan interval" has been set is written to the High Performance model QCPU whose first five digits of serial No. are "04012" or later, the step transition watchdog timers cannot be used.
No processing is performed if the step transition watchdog timers are executed.
(2) The step transition watchdog timers are not available for the Basic model QCPU, Universal model QCPU, and LCPU.

(4) The method for using a step transition watch dog timer is shown below.



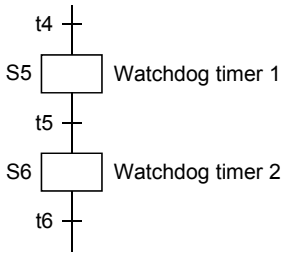
(a) When SM90 is turned ON in the operation output of the step that performs a time check as shown below, the step transition watchdog timer starts timing.

(b) If transition condition a is not satisfied within the set time (10s) after SM90 has turned ON, annunciator F1 turns ON.
(However, the SFC program continues operation.)

(c) When transition condition a is satisfied within the set time and SM90 turns OFF, the step transition watchdog timer stops timing and is reset.

(5) If the annunciators (F0 to F255) turn ON, the number of detected annunciators that turned ON and the annunciator numbers are not stored into SD62, SD63 and SD64 - SD79.

(6) The step transition watchdog timers of the same number can be used at different steps if they do not become active simultaneously.
Example:



As there is no chance that steps 5 and 6 will be concurrently active, the same watch dog timer can be used at both steps.

4 - 86

4.7 SFC Operation Mode Setting

The SFC operation mode setting is used to designate SFC program START conditions, or to designate the processing method at a double START.
Some settings can be made in "SFC setting of PLC parameter dialog box" in the system common setting and the others can be made in "block parameter" of the SFC program.
The SFC operation mode setting items and the resulting operations are shown below.

Item	Description	Setting Range	Default Value	Basic Model QCPU	High Performance Model QCPU, Process CPU Redundant CPU	Universal model; QCPU, LCPU
SFC program start mode	• Designates an “Initial start” or “Resume start” when the SFC program is started.	Initial start/Resume start	Initial start	○	○	○
Start conditions	• Designates whether block 0 is to be started automatically.	Autostart block 0/Do not autostart block 0	Autostart block 0	○	○	○
Output mode when the block is stopped	• Designates the coil output mode at a block STOP.	Turn OFF/Keep ON	Turn OFF	○	○	○
Periodic execution block setting	• Designates the first block No. of the periodic execution blocks.	0 to 319	No setting	×	○	×
	• Designates the time interval for execution of the periodic execution blocks.	1 to 65535 ms				
Act at block multi-activated	• Designates the operation which occurs when a START request is made for a block which is already active.	Stop blocks (a block range can be designated for the stop blocks setting)	Waiting blocks	× (Wait only)	○	× (Wait only)
Act at step multi-activated	• Designates the operation which occurs when a transition (follow-up) is executed to a step which is already active, or when an active step is started.	Waiting blocks/stop blocks (a step range can be designated for the stop blocks or “Waiting blocks” setting)	Transfer	× (Transfer only)	○	× (Wait only)

○: Can be set, ×: Cannot be set.

4 - 87

4.7.1 SFC program start mode

The SFC program start mode setting determines whether an SFC program START (SM321 OFF → ON) is executed by an “Initial start,” or by a Resume start from the preceding execution status.

(1) Settings and corresponding operations

Set whether "initial start" or "resume start" will be selected for the SFC program.

(a) Initial start

The program is started after the active status at a previous stop is cleared.
The operation after a start is performed according to the setting of block 0 START condition.

(b) Resume start

The program is started with the active status at a previous stop (ON to OFF of SM321 or RUN to STOP of CPU module) held.

The SFC program start mode changes depending on the combination of the setting of the "SFC program start mode" in the PLC parameter dialog box and the ON/OFF status of the "special relay for setting SFC program start status (SM322)" as indicated below.

SFC Program Start Mode / Operation	Initial Start		Resume Start	
	SM322: OFF (Initial status) ＊1	SM322: ON (When changed by user)	SM322: OFF (Initial status) ＊1	SM322: ON (When changed by user)
SM321: Turned ON	Initial	Initial	Resume	Initial
Programmable controller: Powered ON			Resume/Initial ＊3	Initial
Programmable controller: Powered OFF and then ON after SM321 is switched from ON to OFF or the CPU module is switched from RUN to STOP			Resume ＊2	Initial
CPU module: Reset and RUN			Resume/Initial ＊6	Initial
CPU module: Reset and RUN after SM321 is switched from ON to OFF or the CPU module is switched from RUN to STOP			Resume ＊2	Initial
CPU module: Switched from STOP to RUN	Resume			
CPU module: STOP, write a program, and then RUN	Initial			

Initial: Initial start, Resume: Resume start

＊1: SM322 is turned ON/OFF by the system according to the setting of the "SFC program start mode" in the PLC parameter dialog box when the CPU module switches from STOP to RUN.

- At initial start setting: OFF
- At resume start setting: ON

＊2: Operation at resume start

At a resume start, the SFC program stop position is held but the status of each device used for the operation output is not held.
Therefore, make latch setting for the devices whose statuses must be held in making a resume start.

- The held coil HOLD step SC becomes inactive, and is not kept held.
  In the Basic model QCPU, Universal model QCPU, and LCPU, the held coil HOLD step SC restarts in the held status under conditions other than those that turns OFF the coil HOLD step, such as turning ON and OFF of SM321 or operating status change (RUN to STOP) of the CPU module.

*3: Depending on the timing, a resume start is disabled and an initial start may be made.
To perform a resume start, turn ON and then OFF SM321 or switch the CPU module from RUN to STOP, and power OFF and then ON the programmable controller.
Note that the Basic model QCPU and the Universal model QCPU with serial number "11042" (first five digits) or earlier always perform an initial start.

*4: A resume start may be made depending on the SFC program change.
If a resume start is made as-is, a start is made from the old step number, leading to a malfunction of the mechanical system.
When any SFC program change (SFC diagram correction such as step addition and deletion) has been made, make an initial start once and then return it to a resume start.
An initial start is always performed in the Basic model QCPU and the Universal model QCPU with serial number (first five digits) "11042" or earlier.

*5: In the Universal model QCPU and LCPU, a resume start is performed if data other than SFC programs are changed.

*6: The Basic model QCPU and Universal model QCPU of which the first 5 digits of the serial number are "11042" always makes an initial start.

POINT

(1) When the programmable controller is powered OFF or the CPU module is reset, the intelligent function modules and special function modules are initialized.
When making a resume start, create an initial program for the intelligent function module/special function module in the block that is always active or in the sequence program.

(2) When the programmable controller is powered OFF or the CPU module is reset, values in the devices without a latch setting are cleared.
To hold the values in the SFC information devices, set a latch range.

4.7.2 Block 0 START condition

The block 0 START condition is designed to set whether block 0 will be automatically activated or not at SFC program START (when SM321 turns from OFF to ON).

Use the block 0 START condition when it is desired to specify the START block at SFC program START according to the product type, etc.
"Auto START ON" is useful when block 0 is used as described below.

- Used as a control block
- Used as a preprocessing block
- Used as an always watched block

(1) Settings and corresponding operations
Set block 0 to "Auto START ON" or "Auto START OFF".
At SFC program START and END step execution, operations are performed as described below.

Setting	Operation	
	At SFC Program START	At end step execution in block 0
Autostart block 0 (default)	• Block 0 is automatically activated, and is executed from its initial step.	• When the end step is reached, the initial step is automatically activated again.
Do not autostart block 0	• Block 0 is activated by a START request resulting from an SFC control “block START” instruction or a block START step, in the same manner as other blocks.	• When the end step is reached, block 0 is deactivated and waits for another START request to be issued again.

4 - 90

4.7.3 Output mode at block STOP

The "output mode at block STOP" is designed to set whether the coil outputs turned ON by the OUT instruction will be held at the time of a stop (coil output held) or all coil outputs will be forcibly turned OFF (coil output OFF) when the corresponding block is stopped temporarily.
Stop the corresponding block temporarily using the "stop RESTART bit" of the SFC information devices or the "block STOP instruction (PAUSE BLm)" of the SFC control instructions.

(1) Settings and corresponding operations

Set the output mode at block STOP in the "output mode at block STOP in PLC parameter dialog box" or the "special register for setting operation output at block STOP (SM325)".
The operation of the SFC program changes depending on the combination of the "output mode at block STOP in PLC parameter dialog box" setting and the SM325 setting.

<table>
<tr><th rowspan="3">Setting of Output Mode at Block Stop in PLC Parameter</th><th rowspan="3">Operation Output at Block Stop (SM325)</th><th rowspan="3">Status of STOP-time Mode Bit</th><th colspan="4">Operation</th></tr>
<tr><th rowspan="2">Active step other than held step (including HOLD step (SC, SE, ST) whose transition condition is not satisfied)</th><th colspan="3">Held step *</th></tr>
<tr><th>Coil HOLD step (SC)</th><th>Operation HOLD step (without transition check) (SE)</th><th>Operation HOLD step (with transition check) (ST)</th></tr>
<tr><td rowspan="2">• Turns OFF (coil output OFF)
• Remains ON (coil output held)</td><td rowspan="2">• OFF (coil output OFF)</td><td>OFF or no setting (immediate stop)</td><td>• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status remains active.</td><td rowspan="2">• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status becomes inactive.</td><td colspan="2" rowspan="2">• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status remains active.</td></tr>
<tr><td>ON (STOP after transition)</td><td>• Normal operation is performed until the transition condition is satisfied.
• When the transition condition is satisfied, the end processing of the corresponding step is performed.
At the same time, the transition destination step becomes active and the block stops immediately.</td></tr>
<tr><td rowspan="2">• Remains ON (coil output held)</td><td rowspan="2">• ON (coil output held)</td><td>OFF or no setting (immediate stop)</td><td>• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held.
• The status remains active.</td><td colspan="3" rowspan="2">• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held.
• The status remains active.</td></tr>
<tr><td>ON (STOP after transition)</td><td>• Normal operation is performed until the transition condition is satisfied.
• When the transition condition is satisfied, the end processing of the corresponding step is performed.
At the same time, the transition destination step becomes active and the block stops immediately.</td></tr>
</table>

*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

(a) Output mode at block STOP in PLC parameter dialog box

Set the status of the output mode at block STOP when the programmable controller is powered ON or the CPU module is reset.

(b) SM325

- For the Universal model QCPU and LCPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) when the CPU module is powered ON or is reset.

Parameter setting	SM325
Turns OFF (coil output OFF)	OFF
Remain ON (coil output held)	ON

2) The output mode at block stop can be changed by turning ON/OFF SM325 during the operation of the SFC program.
(During the operation of the SFC program, the parameter setting is ignored.)

4 - 91

4.7.4 Periodic execution block setting

The periodic execution block setting designates the execution of a given block at specified time intervals rather than at each scan.

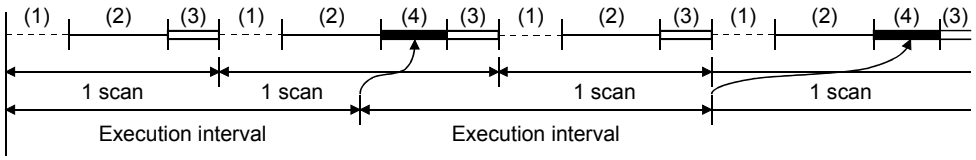
(1) Setting items

Designate the first block number and the time of execution for the periodic execution blocks. When these settings are designated, the “first block” and all subsequent blocks will become periodic execution blocks.

The execution time interval setting can be designated in 1 ms units within a 1 to 65535 ms range.

(2) Periodic execution block operation method

Periodic execution block operation occurs as shown below.



(1) Sequence programs executed at each scan
(2) Blocks executed at each scan
(3) END processing
(4) Periodic execution blocks

1) Until the specified time interval elapses, only the sequence programs and blocks designated for execution at each scan will be executed.

2) When the specified time interval elapses, the periodic execution blocks will be executed following execution of blocks designated for execution at each scan. If the specified time interval is shorter than the scan time, the periodic execution blocks will be executed at each scan in the same manner as the other blocks.

3) The specified time interval countdown is executed in a continuous manner.

POINT

(1) When the parameter where the "High speed interrupt I49 fixed scan interval" has been set is written to the High Performance model QCPU whose first five digits of serial No. are "04012" or later, the fixed-cycle execution block setting cannot be used.
If the fixed-cycle execution block setting is made, no processing is performed and the block remains unchanged from the every scan execution block.

(2) To execute the periodic execution block, the block to be executed periodically must be activated.

(3) The fixed-cycle execution block setting is not available for the Basic model QCPU, Universal model QCPU, and LCPU.

4 - 92

4.7.5 Operation mode at double block START

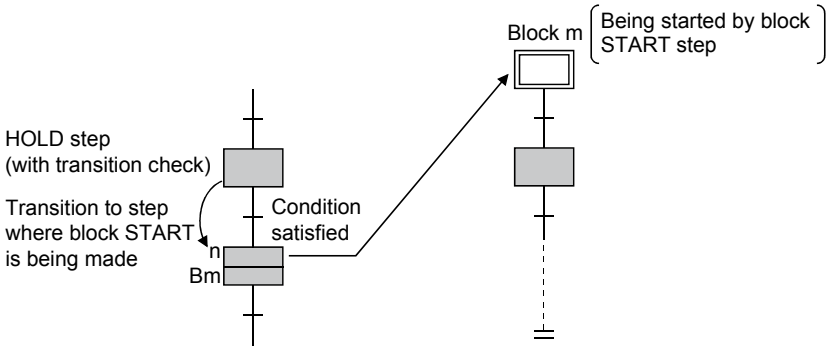
This mode setting designates the operation mode which is to be effective when a block START request occurs (by block START step (m, m)) for a block which is already started.

(1) Settings and corresponding operations

Set the operation mode at block double START to either STOP or WAIT in the "block parameter" of the SFC setting dialog box in the Tools menu.

The operations resulting from these settings are shown below.

Setting	Operation	Remarks
STOP	• A CPU module operation error (BLOCK EXE.ERROR) occurs, and CPU module operation is stopped. • All “Y” outputs switch OFF.	• A block range can be designated for the STOP setting.
WAIT (default)	• CPU module operation continues, and a WAIT status is established when the transition condition is satisfied. The WAIT status continues until the START destination block is deactivated. • A step transition occurs when the START destination block is deactivated, and that block is then reactivated. • If a transition WAIT occurs, the previous step is deactivated, the output is switched OFF, and the operation output will not be executed.	



POINT

(1) When a START request is issued to the block that is already active by execution of the following, the START request is ignored and the processing of the SFC program is continued as is.
- Block START instruction (SET BLm) of SFC control instructions
- ON of Block START/END bit of SFC information devices

(2) For the Basic model QCPU, Universal model QCPU, and LCPU, setting of the operation mode at block double START is not allowed.
The operation mode at block double START is fixed to the default, "WAIT" for them.

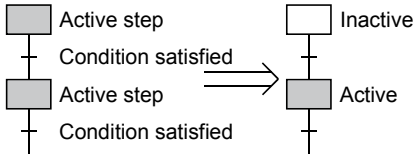
4 - 93

4.7.6 Operation mode at transition to active step (double step START)

This mode setting designates the operation mode which is to be effective when a follow-up function such as an operation HOLD step (with transition check) is used to execute a transition to a step which is already active.

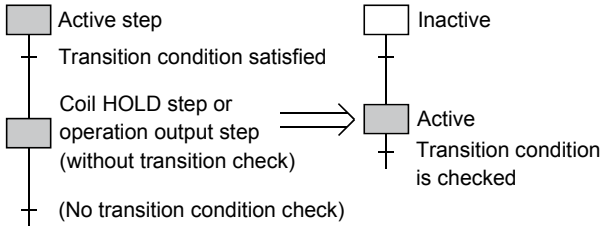
(1) Settings and corresponding operations

For a transition to an active step, set any of STOP, WAIT and TRANSFER in the "block parameter" of the FC setting dialog box in the Tools menu.
The operations resulting from these settings are shown below.

Setting	Operation	Remarks
STOP	• A CPU module operation error (BLOCK EXE.ERROR) occurs, and CPU module operation is stopped. • All “Y” outputs switch OFF.	• A step range can be designated for the STOP setting.
WAIT	• CPU module operation continues, and a WAIT status is established when the transition condition is satisfied. The WAIT status continues until the START destination step is deactivated. • If a transition WAIT occurs, the previous step is deactivated, the output is switched OFF, and the operation output will not be executed.	• A step range can be designated for the WAIT setting.
TRANSFER (default)	• CPU module operation continues, the transition occurs, and the previous step is deactivated and absorbed by the transition destination step. 	

(2) Transition to HOLD step by double START

The following table shows the transition procedure for transitions to coil HOLD steps, operation HOLD steps (with transition check), and operation HOLD steps (without transition check) which occur when the double START condition is satisfied. These transitions occur without regard to the settings described at item (1) above.

Setting	Operation	Remarks
STOP, WAIT, TRANSFER	• The TRANSFER setting applies to all operations, regardless of the setting. • At coil HOLD steps ..... The operation output is restarted, and a transition condition check begins. The PLS instruction for which the input conditions have already been established is non-executable until the input conditions are turned on again. • At operation HOLD steps (without transition check) ..... A transition condition check begins. • At operation HOLD steps (with transition check) ..... Operation continues as is. 	• Following the double START, execution of all subsequent steps where transition conditions are satisfied will occur according to the step attributes.

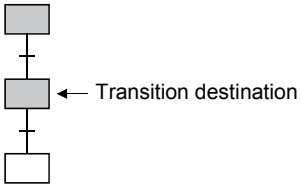
4 - 94

(3) Operation at double START

(a) When transition destination is serial transition

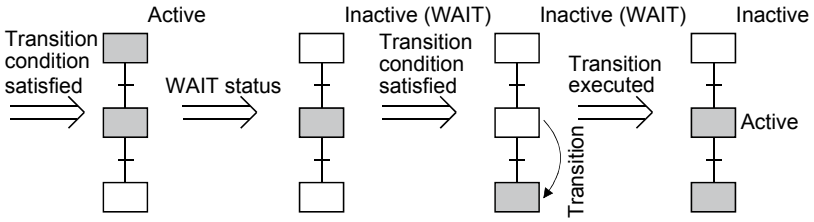
1) When setting is "STOP"

.....If the transition destination is active, an error occurs and the processing of the CPU module stops.



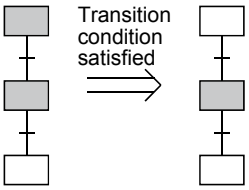
2) When setting is "WAIT"

.....Execution waits until the transition destination step becomes inactive.
When the transition destination step becomes inactive, a transition is executed and the transition destination step becomes active.
In a WAIT status, the previous step is deactivated.



3) When setting is "TRANSFER"

.....A transition is executed and the previous step becomes inactive.

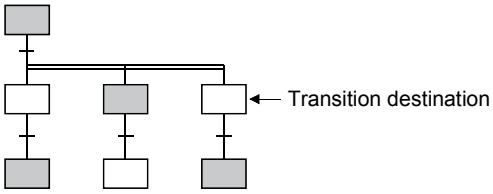


4 - 95

(b) When transition destination is parallel branch

1) When setting is "STOP"

.....If any one of the transition destinations of the parallel branch is active, an error occurs and the processing of the CPU module stops.

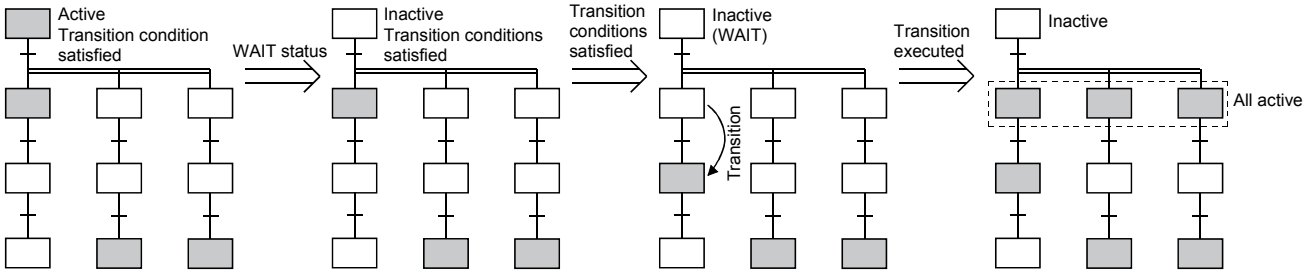


2) When setting is "WAIT"

.....Execution waits until all the transition destination steps of the parallel branch become inactive.

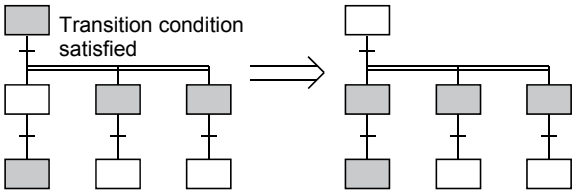
When the transition destination steps all become inactive, a transition is executed and all the first steps of the parallel branch become active.

In a WAIT status, the previous step is deactivated.



3) When setting is "TRANSFER"

.....When any one of the transition destination steps of the parallel branch is active, a transition is executed and the previous step becomes inactive.



REMARKS

When the transition destination steps are all inactive, normal transition processing is performed and all the transition destination steps become active.

POINTS

(1) The operation mode for transition to active step (at step double START) applies to a transition to be executed when a transition condition is satisfied or to a forced transition set using the transition control instruction (SET TRn) of the SFC control instructions.
When the step control instruction (SET Sn) of the SFC control instructions is used to issue a START request to the step that is already active, the request is ignored and the processing continues.

(2) For the Basic model QCPU, Universal model QCPU, and LCPU, setting of the transition to active step (at step double START) is not allowed.
The transition to active step (at step double start) is fixed to the default, "Transition" for them.

4.8 SFC Comment Readout Instruction

SFC comment readout instruction can read comments of steps being activated in the specified blocks or those of the transition condition associated with active steps.
The instructions to read SFC comment are listed below.

Name	Ladder Expression	Function
Instruction to read SFC step comment	S.SFCSCOMR SP.SFCSCOMR	Reads comment of an active step in the specified block.
Instruction to read comment of SFC transition condition	S.SFCTCOMR SP.SFCTCOMR	Reads comment of transition condition associated with an active step in the specified block.

4 - 97

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	×	△*1	△*3	△*2	△*2	×

*1: The serial number (first five digits) shall be 07012 or later.
*2: The serial number (first five digits) shall be 07032 or later.
*3: The serial number (first five digits) shall be 12052 or later, except for the Q00U(J)/Q01U/Q02UCPU.

4.8.1 SFC comment readout instruction (S(P). SFCSCOMR)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J \		Intelligent Function Module U \G	Index Z	Constant K, H	Expansion SFC BLm\Sn	Other Sn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
n1	—	○		—				○	—		BIN16	○	○	—	○	—	—
ⓓ1	—	△*3		—				—	—								
n2	—	○		—				○	—								
n3	—	○		—				○	—								
ⓓ2	△*3	—		—				—	—		Bit						

*3: Local device cannot be used.

S.SFCSCOMR n1 ⓓ1 n2 n3 ⓓ2

SP.SFCSCOMR n1 ⓓ1 n2 n3 ⓓ2

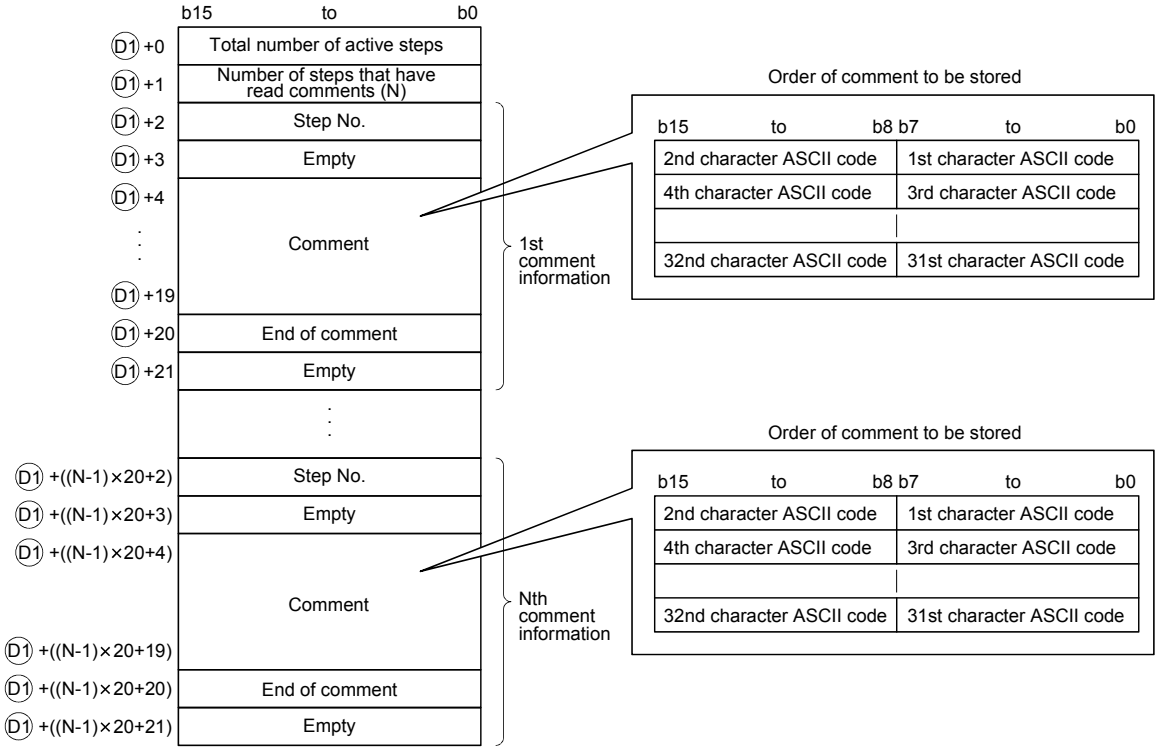
[Set Data]

Set Data	Meaning	Range
n1	Indicates block No. of an SFC program that read comments or device number where block No. is stored.	0 to 319
ⓓ1	Indicates the first number of device that stores comment read.*6	—
n2	Indicates the device number where the number of comments to read or the number of comments is stored.	0 to 256*4
n3	Indicates the number of comments to read in a single scan or device number where the number of comments is stored.	0 to 256*5
ⓓ2	Indicates a device that turns ON for 1 scan at completion of the instruction.	—

*4: When specifying 0, it is processed as 256.
*5: When specifying 0, it is processed as 1.

4 - 98

*6: Comments to be read are stored as follows



Area name	Data to be stored
Total number of steps	• 0000H is stored at S(P). SFCSCOMR instruction, and the total number of steps are stored at completion of comment readout.
Number of steps that have read comments (N)	• 0000H is stored at S(P). SFCSCOMR instruction, and the total number of steps that have actually read comments are stored.
Step No.	• Active step No. that has read comment is stored.
Comment	• Comments that have been read are stored. • Comment area is fixed by a maximum of 32 characters. • In case the word length to be set for 1 comment *7 at the comment range setting is set by 32 or less, 0000H is stored to the area after the number of characters for 1 comment.
End of comment	• 0000H is stored.
Empty	• Not used area (0000H is stored).

*7: The number of characters for each comment in the comment range setting is set in the programming tool.
For details, refer to the manual for the programming tool.

With S(P) .SFCSCOMR instruction, the points calculated by the following formula are occupied from the device No. specified at (D1).

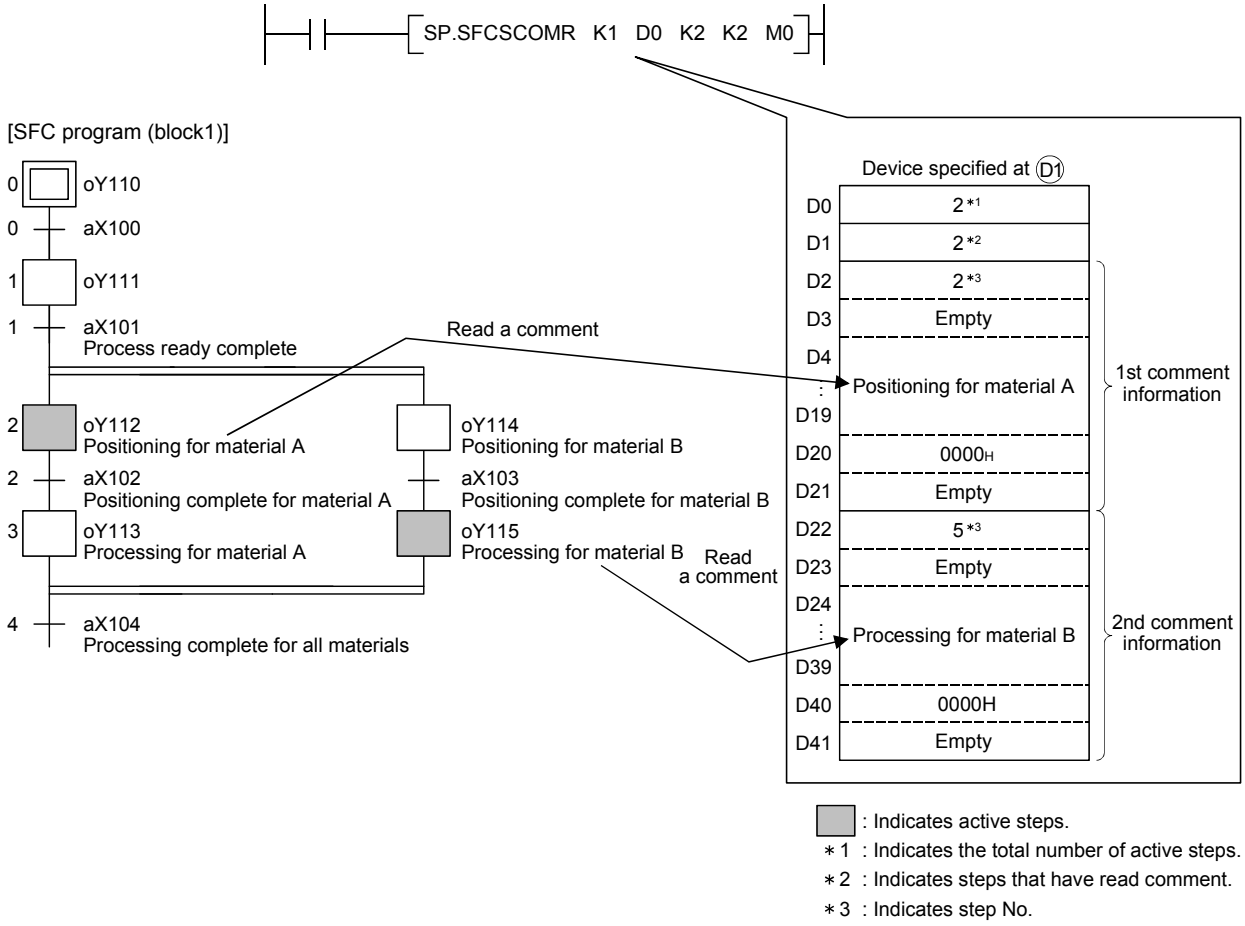
(Points to be used for storing a comment) = 2 + 20 × (number of comment to read (n2))

For (D1), make sure to set device No. that can store the above points successively.

4 - 99

[Functions]

(1) This function reads step comments being activated in the SFC block specified at n1, by the number of comment specified at n2, and stores those to the device number of after specified at (D1).



(2) Executing S(P).SFCSCOMR instruction, SM735 of the special relay (SFC comment readout instruction executing flag) turns ON.
Confirms whether or not S(P).SFCSCOMR instruction is executed by SM735.

(3) In case comments are not set into active steps, "2DH(-)" is stored to the comment area (word length of 32 characters).

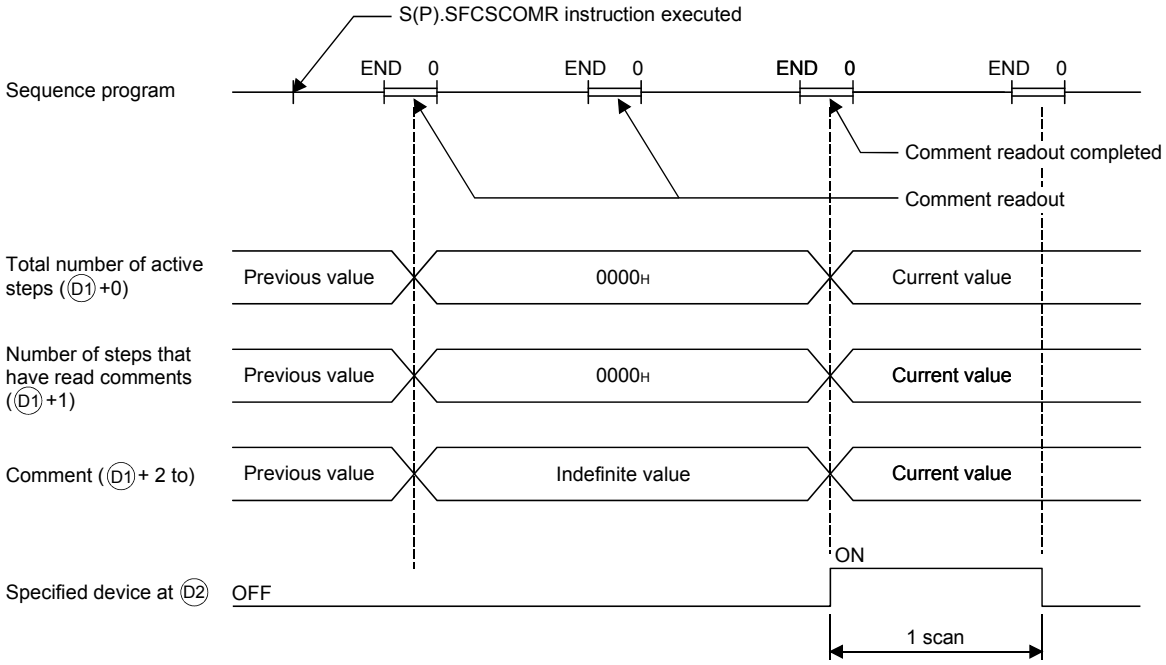
(4) Read comments are stored in ascending order of the step No.

(5) Comments are read from the comment file specified when S(P).SFCSCOMR instruction is executed.

(6) Comments to be read with S(P). SFCSCOMR instruction are those of steps* being activated when executing S(P).SFCSCOMR instruction.

*: As steps retaining coil outputs are not active steps, reading comments is not enabled.

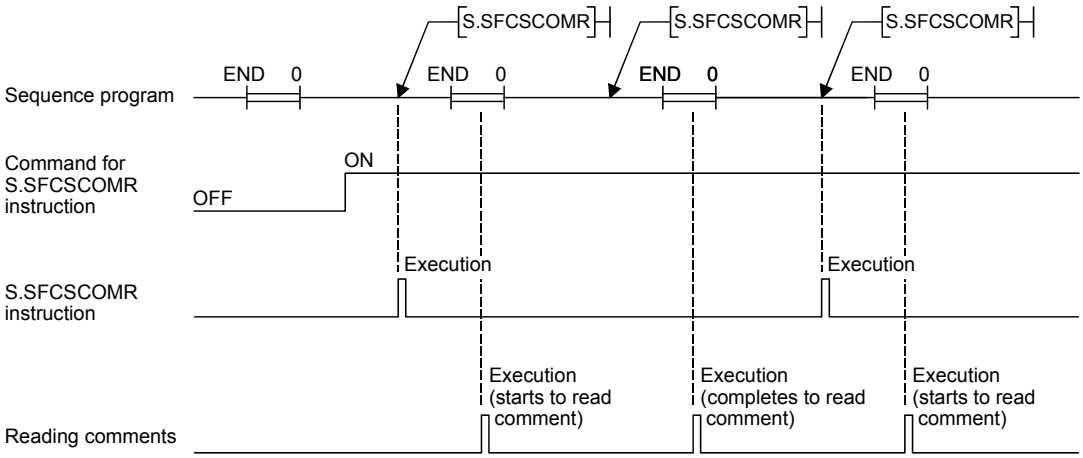
(7) Reading comment is performed at END processing for a scan that has executed S(P).SFCSCOMR instruction.
With per END processing, this function reads the number of comments specified at the number of comments in a single scan (n3).
Comments that are not read in per END processing are followed to the next scan.
Reading comments for active steps (maximum: the number specified at n2) is completed, the device specified at (D2) turns ON for 1 scan.



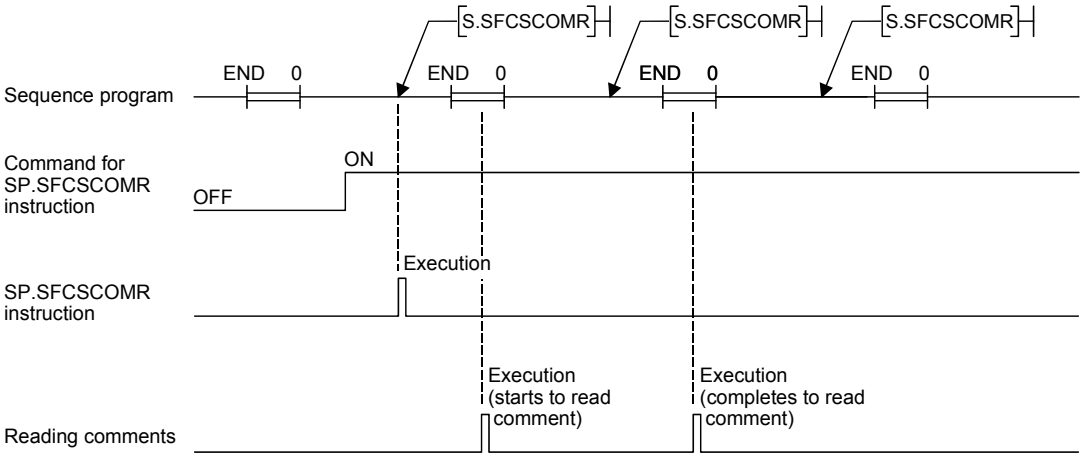
4 - 101

(8) The operation when a command of S(P).SFCSCOMR instruction is in ON status at S(P).SFCSCOMR instruction execution completed is as follows.

(a) S.SFCSCOMR instruction re-executes when a command for S.SFCSCOMR instruction is in ON status.



(b) Even if a command for SP.SFCSCOMR instruction turns ON, SP.SFCSCOMR instruction is not executed.



(9) For the comment files to be used with S(P).SFCSCOMR, set them in the PLC File tab of the PLC parameter dialog box or at "file set instruction (QCDSET(P)) for comments".
Executing S(P). SFCSCOMR without setting the comment file to use, 0 is stored to "the total number of steps (Ⓓ1 +0)" and "the number of steps that have read comments (Ⓓ1 +1)".
At this time, the device specified in Ⓓ2 turns ON for 1 scan.

(10) With S(P).SFCSCOMR instruction, comments stored in the following memories can be read.
- SRAM card (drive 1)
- Flash card (drive 2)
- Standard ROM (drive 4)

The comments stored in the ATA card or SD memory card cannot be read.
If the S(P).SFCSCOMR instruction is executed to the ATA card or SD memory card where the comments are stored, an operation error (error code: 4130) occurs.

(11) While SFC program is not executed, reading comments is not performed even if executing S(P).SFCSCOMR instruction.
Executing S(P).SFCSCOMR instruction at a status without SFC program being executed, 0 is stored to "the total number of steps (Ⓓ1 +0)" and "the number of steps that have read comments (Ⓓ1 +1)".
At this time, the device specified in Ⓓ2 turns ON for 1 scan.

(12) With S(P). SFCSCOMR instruction, comments for the normal SFC program can be read. Comments of a SFC program to control program execution are not read.
Executing S(P).SFCSCOMR instruction specifying a SFC program for program execution control, 0 is stored to "the total number of transit conditions (Ⓓ1 +0)" and "the number of steps that have read comments (Ⓓ1 +1)".
At this time, the device specified in Ⓓ2 turns ON for 1 scan.

(13) S(P).SFCSCOMR instruction cannot be executed simultaneously with S(P).SFCSCOMR instruction or S(P).SFCTCOMR instruction.
Executing S(P).SFCSOMR, and if S(P).SFCSCOMR instruction or S(P).SFCTCOMR instruction is executed before reading comments completed, the 2nd instruction will be deactivated.

(14) When the S(P).SFCSCOMR instruction is attempted to be executed while SM721 is on, the instruction will not be executed. However, when the execution condition is met, the instruction will be executed in the next scan.
SM721 turns on in the following operations:
- The S(P).SFCSCOMR instruction or the S(P).SFCTCOMR instruction is executed.
- The COMRD(P), S(P).FWRITE, S(P).FREAD, or SP.DEVST instruction is executed.
- A file in the ATA card, SD memory card, or standard ROM is accessed by the read from PLC or write to PLC function, or by other file access operations. (Universal model QCPU only)

(15) For the Universal model QCPU, when the S(P).SFCSCOMR instruction is attempted to be executed while online change (inactive block) is executed to the SFC block of comment read target, the instruction will not be executed. However, when the execution condition is met, the instruction will be executed in the next scan.

REMARKS

(1) Make sure to use comments to be read with S(P).SFCSCOMR after the device specified at ⓓ2 turns ON.
Comments to be read before the device specified at ⓓ2 turns ON become an indefinite value.

(2) If the number of steps is larger than that of comments (n3) read in a single scan, the active step comments are divided into the number to be read in a single scan.
Counting the total number of steps is also performed with the same comment number (n3) for 1 scan.
In case transition conditions are remained without being counted when reading comments completed, the counting will be continued for the remained.
Because of this, the number of scans calculated in the following formula is required.
(Comments to be actually stored are the same points stored in (ⓓ1+1)

$$\left(\begin{array}{c}\text{The number of scans until S(P).SFCSCOMR}\\ \text{instruction completed}\end{array}\right)^{*} = \left(\begin{array}{c}\text{The total number of steps}\\ (\text{ⓓ1}+0)\end{array}\right) \div \left(\begin{array}{c}\text{The number of comments}\\ \text{to be read at 1 scan (n3)}\end{array}\right)$$

*: It becomes a round-up below the decimal point.

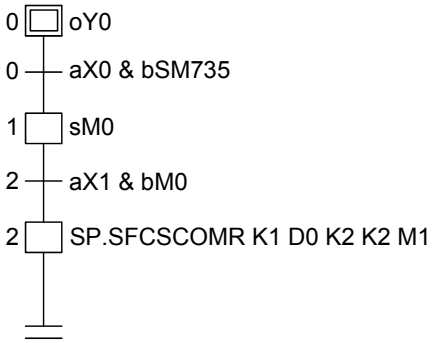
(3) Make sure to perform "batch write of SFC program in RUN status" or "write of comment file in RUN status" with a status of S(P).SFCSCOMR instruction not being executed.
In addition, make sure not to execute S(P).SFCSCOMR during "batch write of SFC program in RUN status" or "write of comment file in RUN status".

[Operation Errors]

- When a comment file specified at execution of S(P).SFCSCOMR instruction does not existed .....Error No. 2410
- When SFC block No. specified at n1 is other than 0 to 319 .....Error No. 4100
- When the number of readout comment specified at n2 is other than 0 to 256 .....Error No. 4100
- When the number of readout comments in a single scan specified at n3 is other than 0 to 256 .....Error No. 4100
- When exceeding the maximum value of the device in which stores comment data to be readout .....Error No. 4101
- When the S(P).SFCSCOMR instruction is executed to the comment file in the ATA card or SD memory card .....Error No. 4130

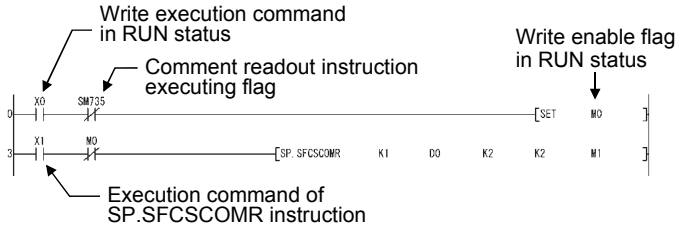
[Program Example]

(1) This program reads 2 comments being activated at the SFC block No.1 when X1 is turned ON, and stores those to the storage device after D0. (The number of comment to be read in a single scan is also set in 2.)
The interlock ladders to perform "batch write of SFC program in RUN status" or "write of comment file in RUN status" are included in the following program.



When step is designated by sequence program

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ANI	SM735
2	SET	M0
3	LD	X1
4	ANI	M0
5	SP. SFCSCOMR	K1 D0 K2 K2 M1

[Procedure for "batch writes of SFC program in RUN status" or "write of comment file in RUN status"]

1) Turns ON the X0 (write execution command in RUN status).
2) M0 (write enable flag in RUN status) is turned ON when SP.SFCSCOMR instruction is deactivated.
3) Turns OFF the X0 (write execution command in RUN status).
4) Performs "batch write of SFC program in RUN status" or "write of comment file in RUN status".
5) Turns OFF the M0 (write enable flag in RUN status) in the device test of the programming tool.
6) SP.SFCTCOMR instruction is executed again when M0 (write enable flag in RUN status) is turned OFF.

Applicable CPU	QCPU					LCPU
	Programmable controller CPU			Process CPU	Redundant CPU	
	Basic	High Performance	Universal			
	×	△*1	△*3	△*2	△*2	×

*1: The serial number (first five digits) shall be 07012 or later.
*2: The serial number (first five digits) shall be 07032 or later.
*3: The serial number (first five digits) shall be 12052 or later, except for the Q00U(J)/Q01U/Q02UCPU.

4.8.2 SFC transition comment readout instruction (S(P). SFCTCOMR)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	Link Direct J \		Intelligent Function Module U \ G	Index Z	Constant K, H	Expansion SFC BLm\Sn	Other Sn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
n1	—	○		—				○	—		BIN16	○	○	—	○	—	—
(D1)	—	△*3		—				—	—								
n2	—	○		—				○	—								
n3	—	○		—				○	—								
(D2)	△*3	—		—				—	—		Bit						

*3: Local device cannot be used.

S.SFCTCOMR n1 (D1) n2 n3 (D2)

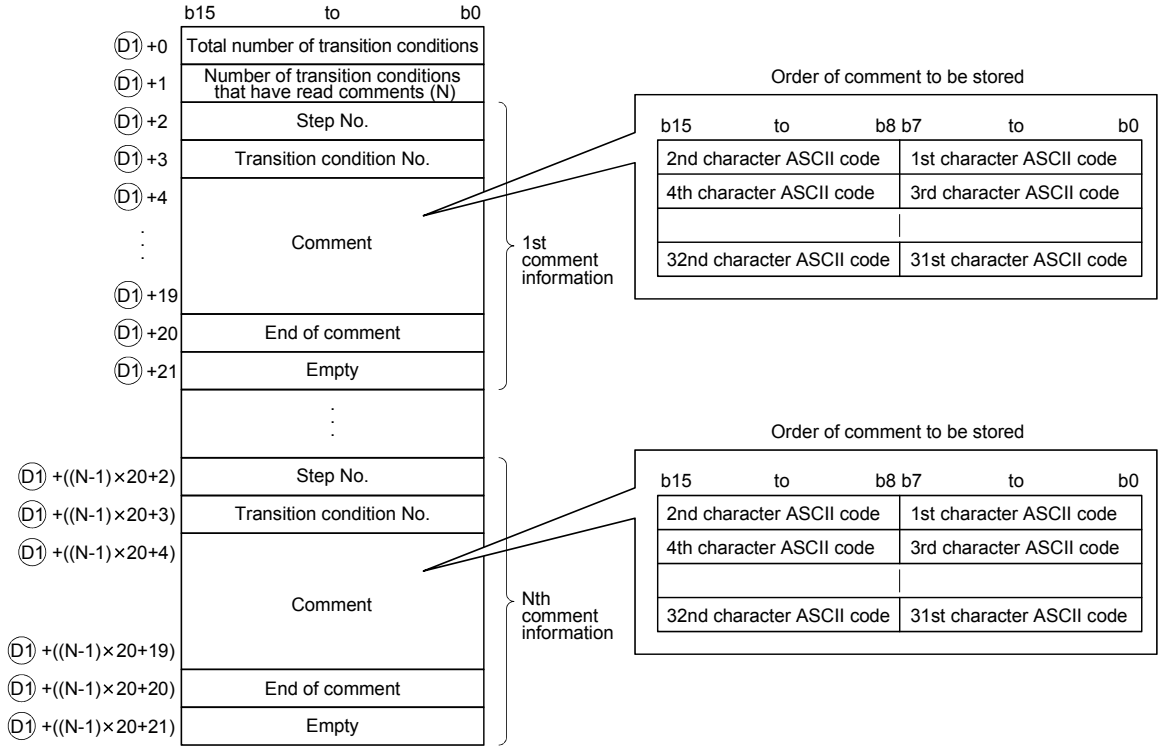
SP.SFCTCOMR n1 (D1) n2 n3 (D2)

[Set Data]

Set Data	Meaning	Range
n1	Indicates block No. of an SFC program that read comments or device number where block No. is stored.	0 to 319
(D1)	Indicates the first number of device that stores comment read. *6	—
n2	Indicates the device number where the number of comments to read or the number of comments is stored.	0 to 256*4
n3	Indicates the number of comments to read in a single scan or device number where the number of comments is stored.	0 to 256*5
(D2)	Indicates a device that turns ON for 1 scan at completion of the instruction.	—

*4: When specifying 0, it is processed as 256.
*5: When specifying 0, it is processed as 1.

*6: Comments to be read are stored as follows.



Area name	Data to be stored
Total number of transition conditions	• 0000H is stored at S(P).SFCTCOMR instruction, and the total number of transition conditions associated with the steps activated when reading comments completed are stored. (Maximum of up to 256 detected)
Number of transition conditions that have read comments (N)	• 0000H is stored at S(P).SFCTCOMR instruction, and the total number of transition condition associated with the active steps that have actually when reading comments completed are stored.
Step No.	• Transition condition step No. that has read comment is stored.
Transition condition No.	• Transition condition No. that has read comment is stored.
Comment	• Comments that have been read are stored. • Comment area is fixed by a maximum of 32 characters. • In case the word length to be set for 1 comment *7 at the comment range setting is set by 32 or less, 0000H is stored to the area after the number of characters for 1 comment.
End of comment	• 0000H is stored
Empty	• Not used area (0000H is stored)

*7: The number of characters for each comment in the comment range setting is set in the programming tool.
For details, refer to the manual for the programming tool.

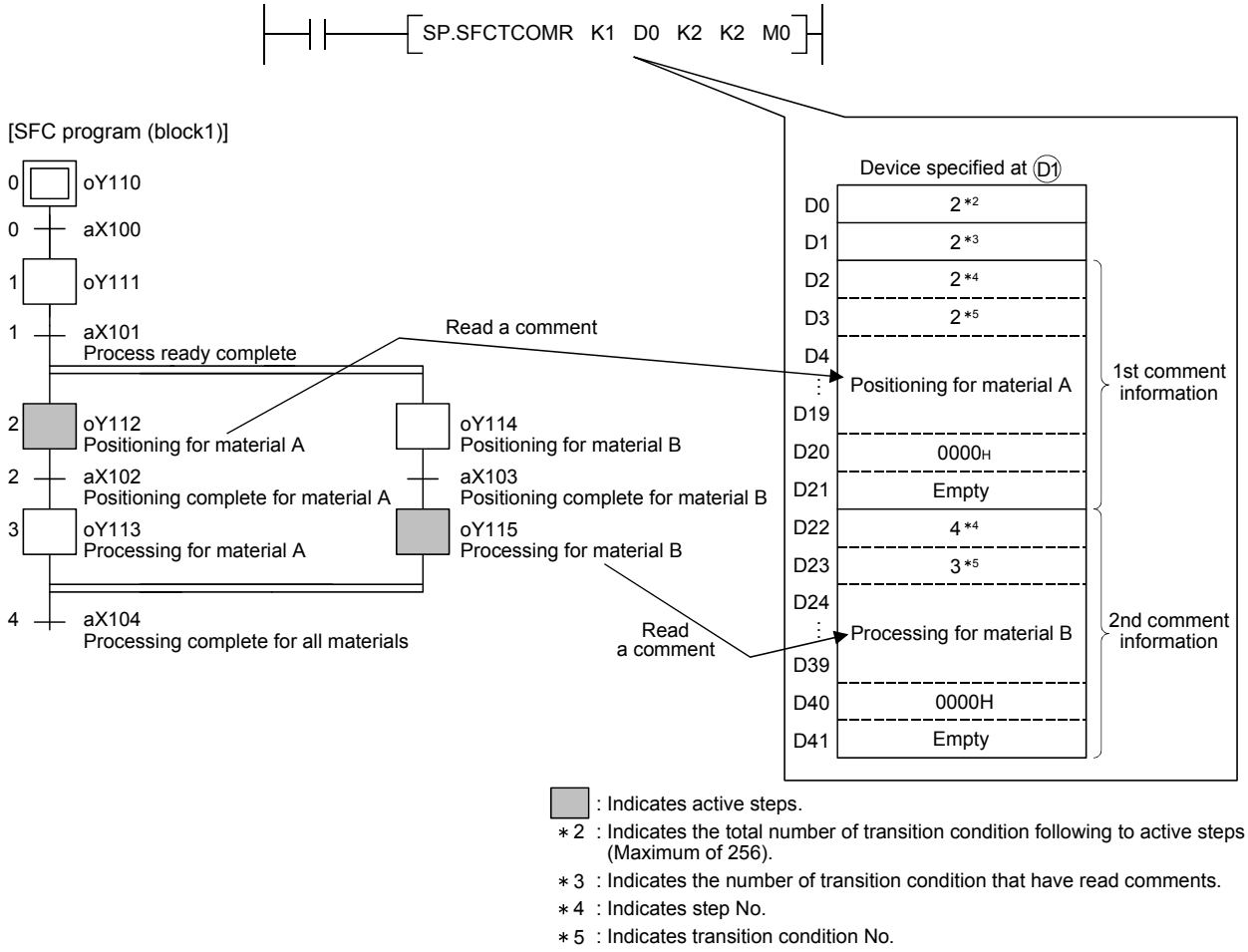
With S(P) .SFCTCOMR instruction, the points calculated by the following formula are occupied from the device No. specified at (D1).

(Points to be used for storing a comment) = 2 + 20 × (number of comment to read (n2))

For (D1), make sure to set device No. that can store the above points successively.

[Functions]

(1) This function reads comments of the transition condition*1 associated with steps activated in the SFC block specified at n1 with the number of comments specified at n2, and stores those to the device number of after specified at (D1).



: Indicates active steps.

*2 : Indicates the total number of transition condition following to active steps (Maximum of 256).

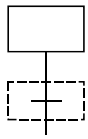
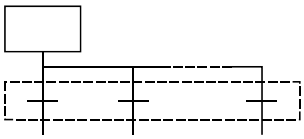
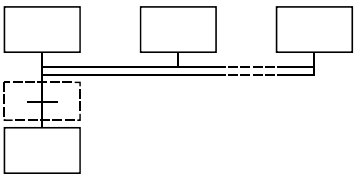
*3 : Indicates the number of transition condition that have read comments.

*4 : Indicates step No.

*5 : Indicates transition condition No.

*1: Transition condition associated with active steps is shown below.

- Serial transition is a transition condition for right under a step.
- Selection branching is a transition condition for all branches.
  Comment of transition condition is read from left to right in the SFC diagram.
- Parallel coupling is a transition condition for after parallel coupling.
  Comments are read only when steps with parallel-coupled are all activated
  Step No. described at the most right edge is stored for transit condition to be read.

Serial transition	Selection branching	Parallel coupling
		

: Indicates a transition condition associated with steps.

4 - 108

(2) Executing S(P).SFCTCOMR instruction, SM735 of the special relay (SFC comment readout instruction executing flag) turns ON.
Confirms whether or not S(P).SFCTCOMR instruction is executed by SM735.

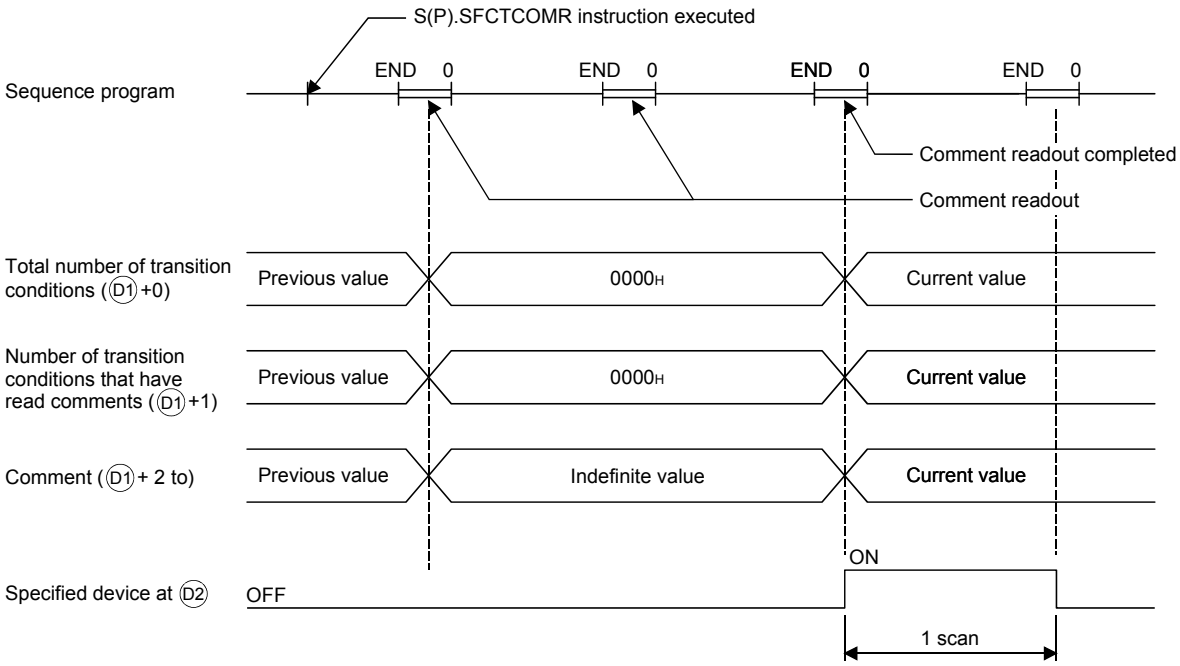
(3) In case comments are not set into active steps, "2DH(-)" is stored to the comment area (word length of 32 characters).

(4) Read comments are stored in ascending order of the step No.

(5) Comments are read from the comment file specified when S(P).SFCTCOMR is executed.

(6) Comments to read with S(P).SFCTCOMR, comments of transition condition associated with active steps of ＊ with when S(P).SFCTCOMR instruction is executed.
Because of this, step comments to be activated after S(P).SFCTCOMR execution can not be read.
＊: As coil retention step at a status of retaining coil output or operation retention step retaining operation condition (without transition check) is not active step, a comment cannot be read.

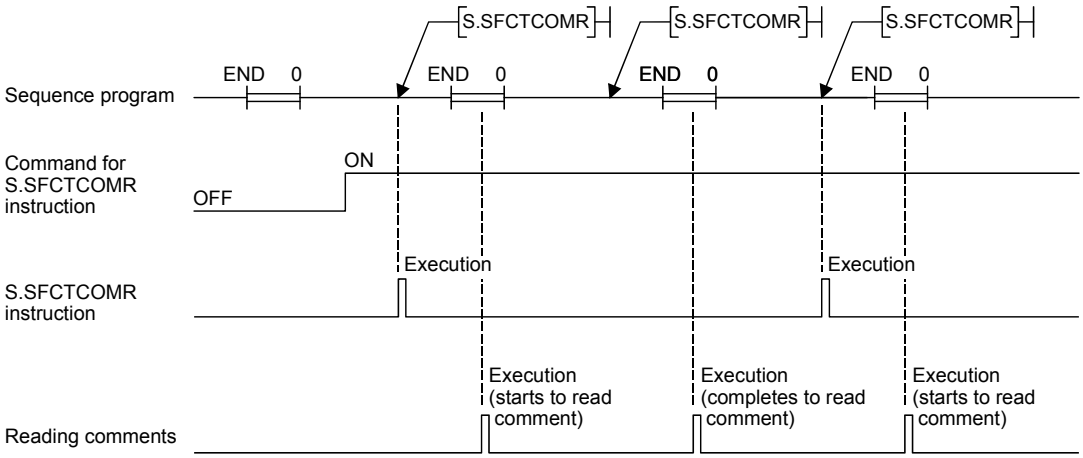
(7) Reading comment is performed at END processing for a scan that has executed S(P).SFCTCOMR instruction.
The number of comments specified at n3 is read per END processing.
Comments that are not read per END processing are followed to the next END processing.
Reading comments for transition conditions (maximum: the number specified at n2) associated with active steps is completed, the device specified at ⓓ2 turns ON for 1 scan.



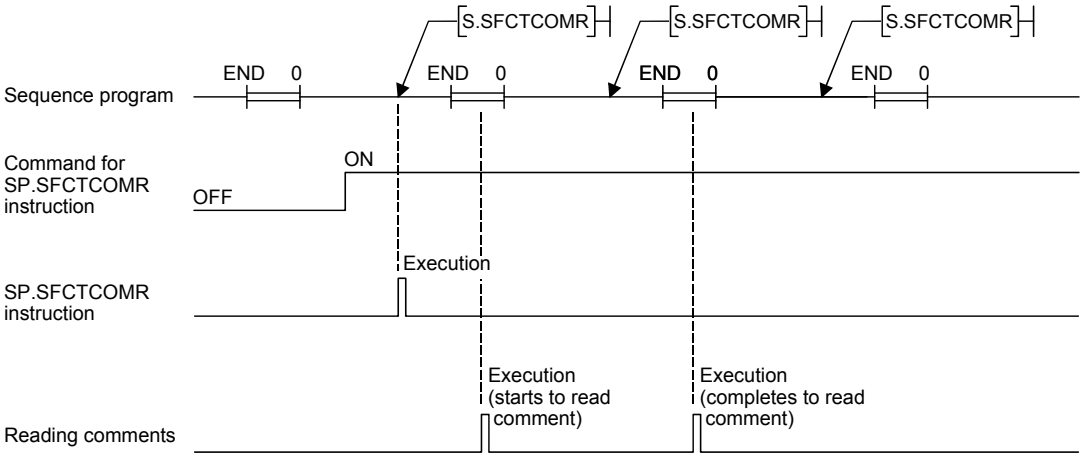
4 - 109

(8) The operation when a command of S(P).SFCTCOMR instruction is in ON status at S(P).SFCTCOMR instruction execution completed is as follows.

(a) S.SFCTCOMR instruction re-executes when a command for S.SFCTCOMR instruction is in ON status.



(b) Even if a command for SP.SFCTCOMR instruction turns ON, SP.SFCTCOMR instruction is not executed.



(9) For the comment files to be used with S(P).SFCTCOMR, set them in the PLC File tab of the PLC parameter dialog box or at "file set instruction (QCDSET(P)) for comments".
Executing S(P).SFCTCOMR without setting of comment file to use, 0 is stored to "the total number of transition conditions (Ⓓ1 +0)" and "the number of transit condition that have read comments(Ⓓ1 +1)".
At this time, the device specified in Ⓓ2 turns ON for 1 scan.

(10) With S(P).SFCTCOMR instruction, comments stored in the following memories can be read.
- SRAM card (drive 1)
- Flash card (drive 2)
- Standard ROM (drive 4)

The comments stored in the ATA card or the SD memory card cannot be read.
If the S(P).SFCSCOMR instruction is executed to the ATA card or SD memory card where the comments are stored, an operation error (error code: 4130) occurs.

(11) While SFC program is not executed, reading comments is not performed even if executing S(P).SFCTCOMR instruction.
Executing S(P).SFCTCOMR at a status of SFC program not being activated, 0 is stored to "total number of transition conditions (Ⓓ1 +0)" and "the number of transition condition that have read comments (Ⓓ1+1)".
At this time, the device specified in Ⓓ2 turns ON for 1 scan.

(12) With S(P). SFCTCOMR instruction, comments for the normal SFC program can be read.
Comments of an SFC program to control program execution are not read.
Executing S(P). SFCTCOMR instruction specifying the SFC program to control execution, 0 is stored to "the total number of transit conditions (Ⓓ1+ 0)" and "the number of transient conditions (Ⓓ1+1) ".
At this time, the device specified in Ⓓ2 turns ON for 1 scan.

(13) S(P).SFCTCOMR instruction cannot be executed simultaneously with S(P).SFCTCOMR instruction or S(P).SFCTCOMR instruction.
Executing S(P).SFCTOMR, and if S(P).SFCSCOMR instruction or S(P).SFCTCOMR instruction is executed before reading comments completed, the 2nd instruction will be deactivated.

(14) When the S(P).SFCTCOMR instruction is attempted to be executed while SM721 is on, the instruction will not be executed. However, when the execution condition is met, the instruction will be executed in the next scan.
SM721 turns on in the following operations:
- The S(P).SFCSCOMR instruction or the S(P).SFCTCOMR instruction is executed.
- The COMRD(P), S(P).FWRITE, S(P).FREAD, or SP.DEVST instruction is executed.
- A file in the ATA card, SD memory card, or standard ROM is accessed by the read from PLC or write to PLC function, or by other file access operations. (Universal model QCPU only)

(15) For the Universal model QCPU, when the S(P).SFCTCOMR instruction is attempted to be executed while online change (inactive block) is executed to the SFC block of comment read target, the instruction will not be executed. However, when the execution condition is met, the instruction will be executed in the next scan.

REMARKS

(1) Make sure to use comments to be read with S(P).SFCTCOMR after the device specified at Ⓓ2 turns ON. Comments to be read before the device specified at Ⓓ2 turns ON become an indefinite value.

(2) If the number of transition conditions associated with active steps is larger than that of comments to be read in a single (n3), the active step comments are divided into the number to be read in a single scan.
Counting the total number of steps is also performed with the same comment number (n3) for 1 scan.
In case transition conditions are remained without being counted when reading comments completed, the counting will be continued for the remained. Because of this, the number of scans calculated in the following formula is required.
(Comments to be actually stored are the same points stored in Ⓓ1 +1)

$$\left(\begin{array}{c}\text{The number of scans until S(P).SFCTCOMR}\\ \text{instruction completed}\end{array}\right)^{*} = \left(\begin{array}{c}\text{Total number of transition}\\ \text{conditions (Ⓓ1 +0)}\end{array}\right) \div \left(\begin{array}{c}\text{The number of comments}\\ \text{to be read at 1 scan (n3)}\end{array}\right)$$

*: It becomes a round-up below the decimal point.

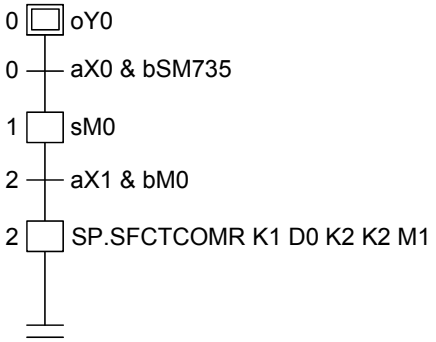
(3) Make sure to perform "batch write of SFC program in RUN status" or "write of comment file in RUN status" with a status of S(P).SFCTCOMR instruction not being executed.
In addition, make sure not to execute S(P).SFCTCOMR during "batch write of SFC program in RUN status" or "write of comment file in RUN status".

[Operation Errors]

- When a comment file specified at execution of S(P).SFCTCOMR instruction does not existed .....Error No. 2410
- When SFC block No. specified at n1 is other than 0 to 319 .....Error No. 4100
- When the number of readout comment specified at n2 is other than 0 to 256 .....Error No. 4100
- When the number of readout comments in a single scan specified at n3 is other than 0 to 256 .....Error No. 4100
- When exceeding the maximum value of the device in which stores comment data to be readout .....Error No. 4101
- When the S(P).SFCTCOMR instruction is executed to the comment file in the ATA card or SD memory card .....Error No. 4130

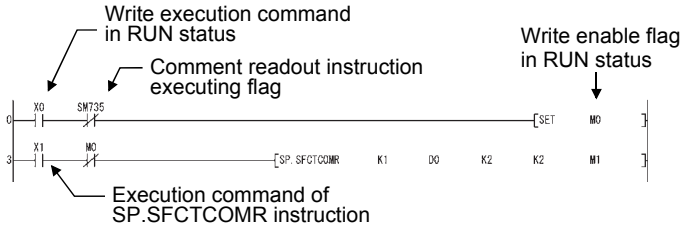
[Program Example]

(1) This program reads 2 comments associated with steps being activated at the SFC block No.1 when X1 is turned ON, and stores those to the storage device after D0. (The number of comment to be read in a single scan is also set in 2.)
The interlock ladders to perform "batch write of SFC program in RUN status" or "write of comment file in RUN status" are included in the following program.



When step is designated by sequence program

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ANI	SM735
2	SET	M0
3	LD	X1
4	ANI	M0
5	SP.SFCTCOMR	K1 D0 K2 K2 M1

[Procedure for "batch writes of SFC program in RUN status" or "write of comment file in RUN status"]

1) Turns ON the X0 (write execution command in RUN status).
2) M0 (write enable flag in RUN status) is turned ON when SP.SFCTCOMR instruction is deactivated.
3) Turns OFF the X0 (write execution command in RUN status).
4) Performs "batch write of SFC program in RUN status" or "write of comment file in RUN status".
5) Turns OFF the M0 (write enable flag in RUN status) in the device test of the programming tool.
6) SP.SFCTCOMR instruction is executed again when M0 (write enable flag in RUN status) is turned OFF.

MEMO

4 - 114

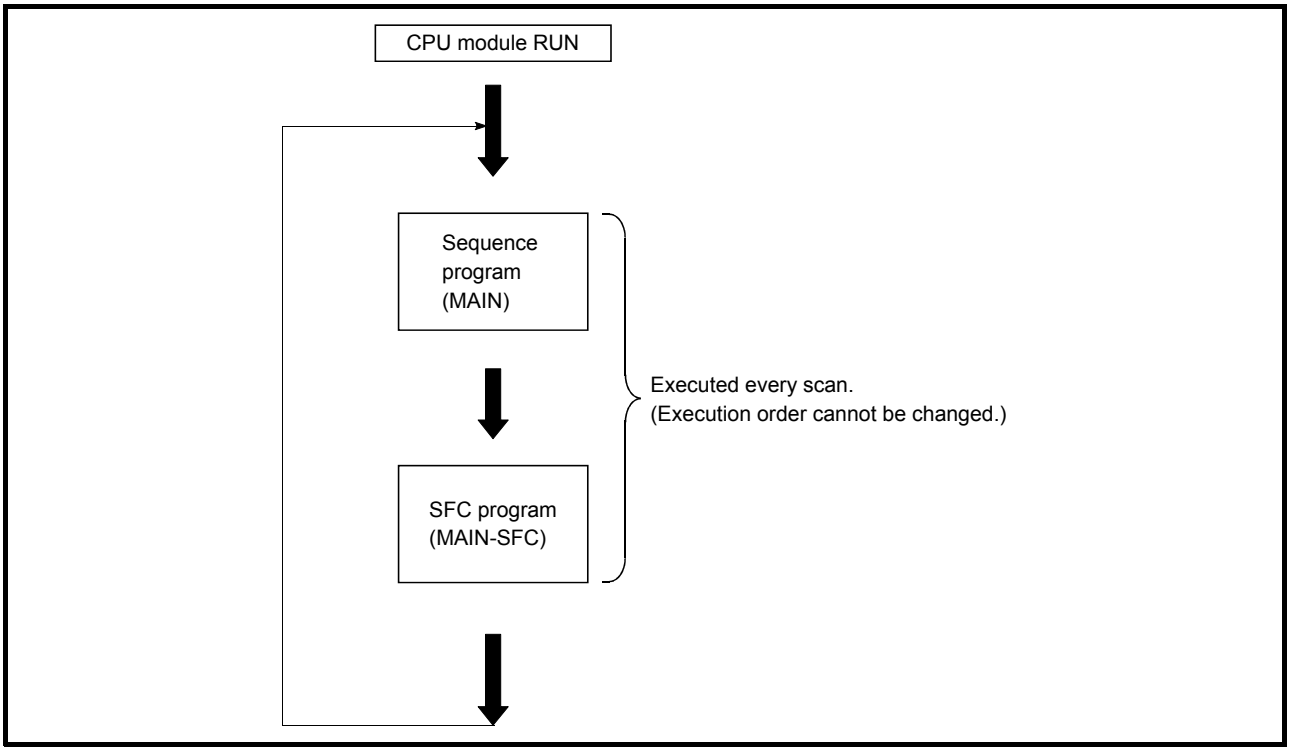
5. SFC PROGRAM PROCESSING SEQUENCE

5.1 Whole Program Processing of Basic Model QCPU

This section explains the program processing of the Basic model QCPU.
Since this manual describes only the outline, refer to the QCPU User's Manual (Function Explanation, Programming Fundamentals) for details.

5.1.1 Whole program processing sequence

The Basic model QCPU can create and execute two programs, "sequence program" and "SFC program", in the program memory.
(Two sequence programs or two SFC programs cannot be created. A SFC program for program execution management cannot be created either.)



(a) The execution types of the sequence program and SFC program are fixed to the "scan execution type".
(The execution types of the sequence program and SFC program are fixed.)

(b) The Basic model QCPU executes the SFC program after execution of the sequence program.
(The execution order of the sequence program and SFC program is fixed.)

(c) The file name of the sequence program is fixed to "MAIN".
Also, the file name of the SFC program is fixed to "MAIN-SFC".

POINT
When both the "sequence program" and "SFC program" exist in the program memory, both programs are executed. Delete the programs, which will not be executed, from the program memory. When ROM operation is performed, delete the programs, which will not be executed, from the standard ROM.

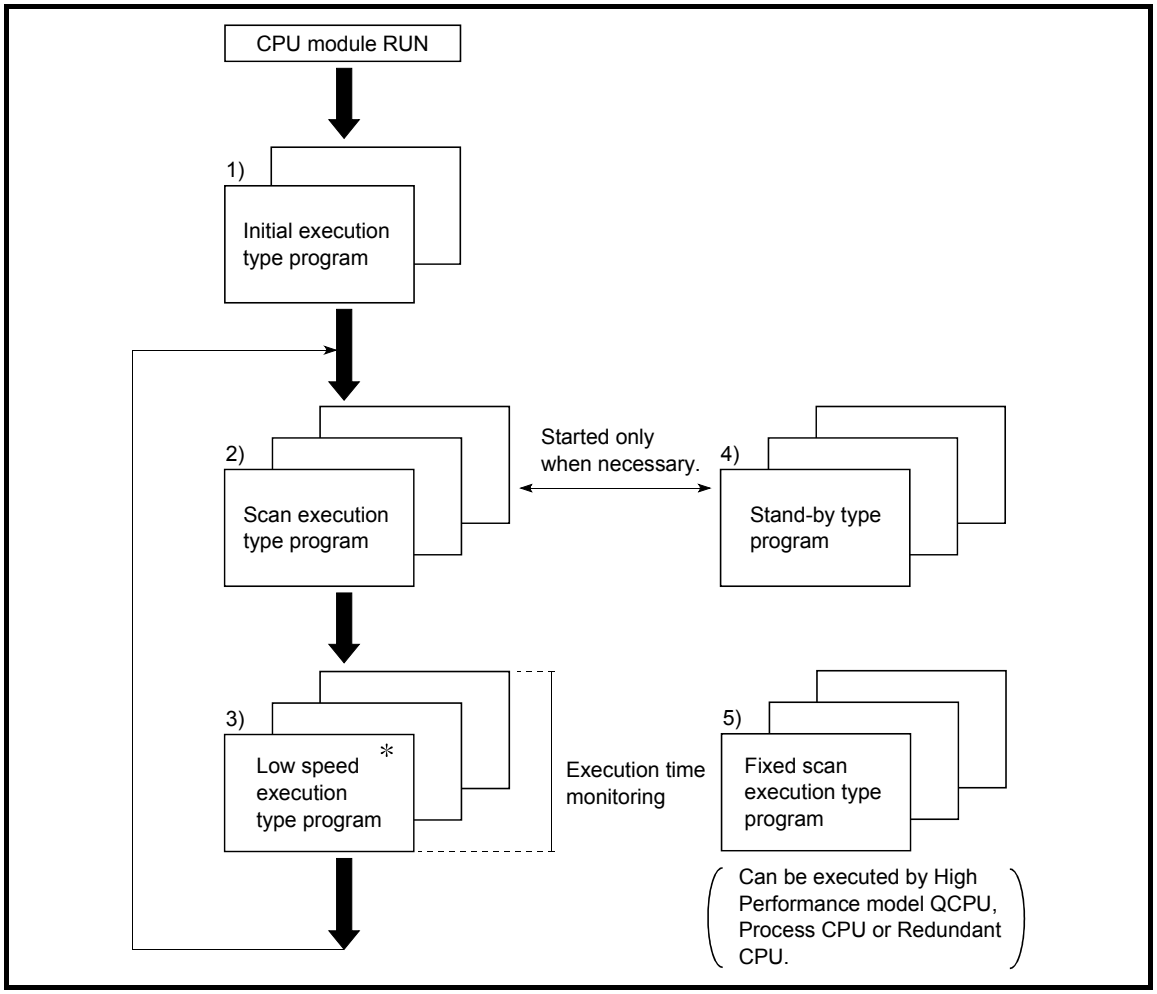
5 - 1

5.2 Whole Program Processing of High Performance Model QCPU, Process CPU, Redundant CPU, Universal Model QCPU, and LCPU

This section explains the whole program processing of the High Performance model QCPU, Process CPU, Redundant CPU, Universal model QCPU, and LCPU.
Since this manual describes only the outline, refer to the QCPU User's Manual (Function Explanation, Programming Fundamentals) for details.

5.2.1 Whole program processing sequence

The High Performance model QCPU, Process CPU, Redundant CPU, Universal model QCPU, and LCPU can store multiple programs in the program memory as files, and can execute multiple files concurrently or the specified file only.
The whole operation image is as shown below.



*: The low-speed execution type program execution is not available for the Redundant CPU, Universal model QCPU, and LCPU.

5 - 2

Execution Type		Description	SFC Compatibility
(1)	Initial execution type program (Initial)	• Executed only in one scan when the programmable controller is powered ON or the CPU module is switched from STOP to RUN. • After that switches to a stand-by type program.	×
(2)	Scan execution type program (Scan)	• Program executed every scan.	Max. 124 programs (changes depending on the CPU module type) • SFC program: Max. 2 programs∗1 • Normal SFC program: 1 program • SFC program for program execution management: 1 program∗2
(3)	Low speed execution type program (Low speed)	• Program executed in the extra time of the constant scan time, or program executed only during preset time.	×
(4)	Stand-by type program (wait)	• Programs such as a subroutine program and interrupt program. • Started by the program START instruction for execution.	Max. 124 programs (changes depending on the CPU module type) SFC program • Normal SFC program: Multiple programs can be set • SFC program for program execution management: Cannot be set
(5)	Fixed scan execution type program (Fixed scan)	• Program executed in a fixed cycle.	×

× : Cannot be set.

∗1: Only one program is allowed for the Universal model QCPU and LCPU.
∗2: The Universal model QCPU and LCPU do not support SFC programs for program execution management.

REMARKS

(1) When the SFC program set as a stand-by type program is to be started, the SFC program in execution must be switched to a stand-by type program before it is started.
Refer to Section 5.2.2 for the method of switching between the scan execution type program and stand-by type program.

(2) Specify the execution type of each program file in "Program" of the PLC parameter dialog box.

(3) In the "Program" of the PLC parameter dialog box, set the normal SFC program to the number higher than that of the SFC program for program execution management.
If the normal SFC program is set to the number lower than that of the SFC program for program execution management, an error may occur when the SFC program set as a stand-by type program is started.

5.2.2 Execution type designation by instructions

The "execution designation by instruction" function enables the execution type set in the program setting of the PLC parameter dialog box to be changed by the instruction.
This function can be applied to normal SFC programs only. (Inapplicable to the SFC programs for program execution management.)
Execution designation by instruction will be explained.

(1) Instructions and corresponding operations

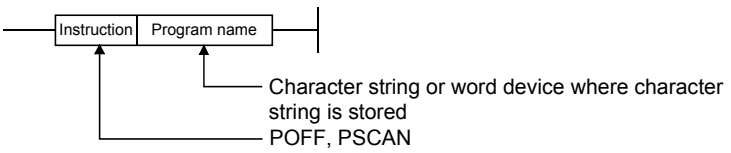
Instruction	Operation	SFC Compatibility
PSTOP	• Switches the program of the specified file name to a standby status, beginning in the next scan.	×
POFF	• Executes the end processing of all blocks in the next scan in the SFC program of the specified file name, and switches the program to a standby status in the second scan after execution of the instruction.	○
PSCAN	• Switches the program of the specified file name to a scan execution type, beginning in the next scan. • The execution order of multiple programs changes depending on the program setting order in the PLC parameter dialog box.	○
PLOW	• Switches the program of the specified file name to a low-speed execution type, beginning in the next scan. • The execution order of multiple programs changes depending on the program setting order in the PLC parameter dialog box.	×

○: Compatible, ×: Incompatible

REMARKS

- The following conditions will result in an operation error:
  - When the specified program does not exist. (error No. 2410).
  - When the PSTOP or PLOW instruction is executed (error No. 2412)
  - When an SFC program is designated by the PSCAN instruction while scanning is in progress at another SFC program (error No. 2412).
  - When the specified SFC program is in scan execution or not can be confirmed with the PCHK instruction.
    (For the Basic model QCPU, Universal model QCPU, and LCPU, the PCHK instruction is not available.)
    For details of the PCHK instruction, refer to the QCPU Programming Manual (Common Instructions).

(2) Instruction format



5 - 4

(3) Processing time required to switch SFC program from WAIT status to scan status

The processing time required to switch an SFC program from a WAIT status to a scan status is shown below.

Although the scanning time is extended by the amount of the processing time, this will not result in a watch dog timer error detection.

No system processing time is required when switching from a scan status to a WAIT status.

Switching time =(number of created programs × Km) + (number of created steps × Kn) + (SFC program capacity × Kp) + Kq

	High Performance model QCPU		Process CPU	Universal model QCPU				
	Q02CPU	QnHCPU	QnPHCPU	Q00UJCPU, Q00UCPU, Q01UCPU	Q02UCPU	Q03UDCPU, Q03UDECPU	Q04UDHCPU, Q06UDHCPU, Q04UDEHCPU, Q06UDEHCPU	Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU
Km	451.9μs	194.7μs	194.7μs	11.9μs	11.2μs	10.6μs	4.4μs	7.3μs
Kn	19.1μs	8.2μs	8.2μs	3.8μs	3.6μs	0.7μs	0.5μs	1.1μs
Kp	6.2μs	2.7μs	2.7μs	0.9μs	0.8μs	0.8μs	0.7μs	0.7μs
Kq	-	-	-	8893.5μs	8470μs	13970μs	8070μs	8100μs

	Universal model QCPU		LCPU			
	Q03UDVCPU	Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, Q26UDVCPU	L02SCPU, L02SCPU-P	L02CPU, L02CPU-P	L06CPU, L06CPU-P	L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT
Km	9.6μs	6.3μs	11.8μs	11.6μs	4.4μs	7.3μs
Kn	0.6μs	1.0μs	3.8μs	2.8μs	0.5μs	1.1μs
Kp	0.7μs	0.6μs	0.9μs	0.8μs	0.7μs	0.7μs
Kq	8920μs	5400μs	8893.5μs	8460.0μs	8070.0μs	8100.0μs

5.2.3 SFC program for program execution management

This SFC program can be used to manage the program execution sequence when multiple program file switching is required.
In addition to a normal SFC program, only one block can be created and executed for a single file of an SFC program for program execution management.

(1) How to create SFC program for program execution management

(a) Number of files and blocks
In addition to a normal SFC program, only one file of an SFC program for program execution management can be created as a scan execution type program.
Only one block of the SFC program for program execution management can be created.

(b) Usable instructions
The SFC diagram symbols (except the block START steps (Bm, Bm)) and steps that can be used in an SFC program and the sequence instructions that can be used in transition conditions can all be used.

POINT

If block start steps (Bm, Bm) are described, a "BLOCK EXE. ERROR" error (error No. 4621) will occur during SFC program execution and the CPU module will stop the execution.

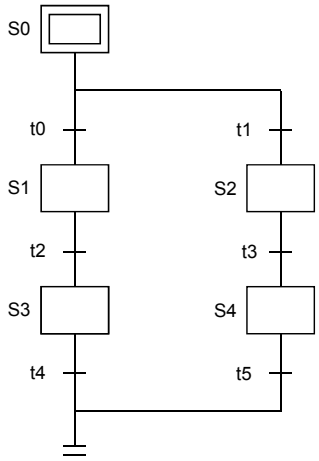
(2) Execution procedure
The program is started automatically when registered as a scan execution type program.
At end step processing, the initial step is reactivated and processing is repeated.

REMARKS

(1) Use the peripheral device to select between the SFC program for program execution management and the normal SFC program.
For details regarding the setting procedure, refer to the GX Developer Operating Manual (MELSAP-L).
(2) Periodic execution block settings (see Section 4.7.4) cannot be defined the SFC programs for program execution control.
If an SFC program for program execution control is set in a periodic execution block, the execution of the SFC program will not be performed.
(3) The Basic model QCPU, Universal model QCPU, and LCPU do not support SFC programs for program execution management.
(4) The SFC program for program execution management cannot be set as a standby type program. In addition, execution designation by POFF or PSCAN instruction cannot be applied to the program.
(5) The SFC control instructions cannot be executed for the SFC program for program execution management. (Refer to Section 4.4.)

(3) Example of program execution management SFC programs

In the following example, SFC program ABC is executed when condition 1 is satisfied, and SFC program XYZ is executed when condition 2 is satisfied.



t0 ---- Condition 1	· When condition 1 is satisfied, execution proceeds to S1.
t1 ---- Condition 2	· When condition 2 is satisfied, execution proceeds to S2.
S1 ---- POFF "XYZ"	· SFC program XYZ is changed into a stand-by type program.
S2 ---- POFF "ABC"	· SFC program ABC is changed into a stand-by type program.
t2 ---- PCHK "XYZ" & INV	· When SFC program XYZ is deactivated, execution proceeds to S3.
t3 ---- PCHK "ABC" & INV	· When SFC program ABC is deactivated, execution proceeds to S4.
S3 ---- PSCAN "ABC"	· SFC program ABC is changed into a scan execution type program.
S4 ---- PSCAN "XYZ"	· SFC program XYZ is changed into a scan execution type program.
t4 ---- PCHK "ABC"	· When SFC program ABC is activated, execution proceeds to the end step.
t5 ---- PCHK "XYZ"	· When SFC program XYZ is activated, execution proceeds to the end step.

5 - 7

5.3 SFC Program Processing Sequence

5.3.1 SFC program execution

The SFC program is executed once per scan.

(1) Basic model QCPU

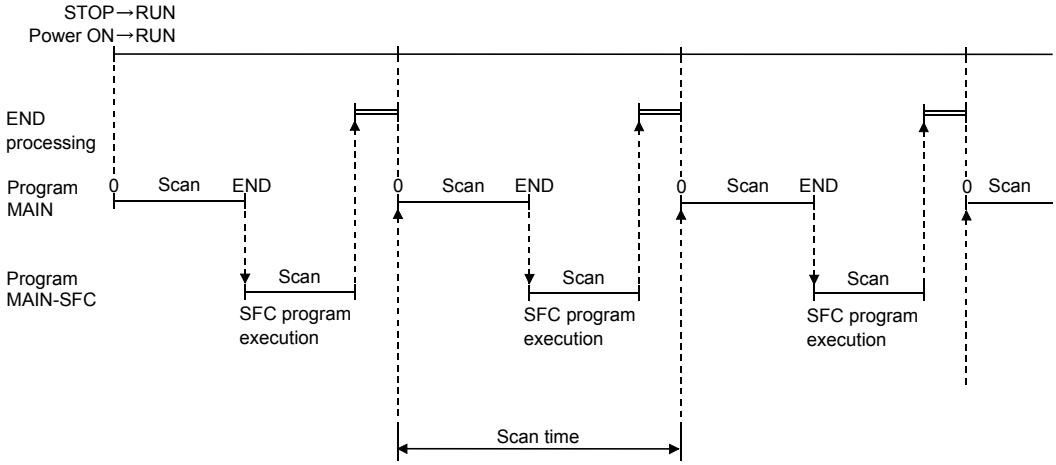
The Basic mode QCPU executes a sequence program and then executes an SFC program.

The program execution status is shown below under the following condition.

[Condition]

1) SFC program: Set to Auto START ON

[Program execution]



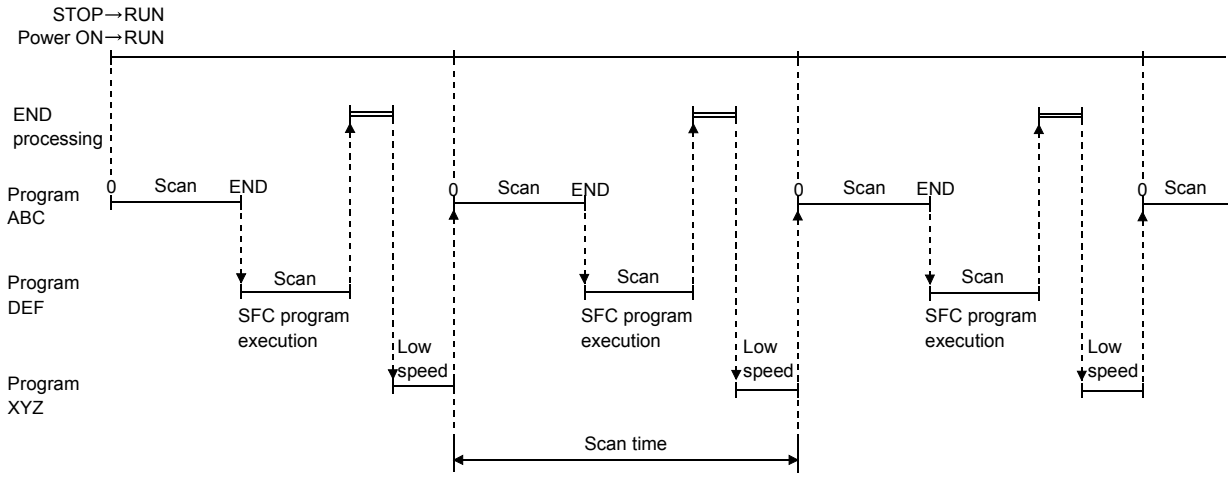
5 - 8

(2) High Performance model QCPU, Process CPU, Redundant CPU, Universal model QCPU, and LCPU

The High Performance model QCPU, Process CPU, Redundant CPU, Universal model QCPU, and LCPU can store multiple programs in the program memory and execute them.
(Scan execution is enabled for two SFC programs (one SFC program for program execution management and one normal SFC program). ∗1
Multiple programs are executed in the order of the program setting in the PLC parameter dialog box.

The execution status of multiple programs is shown below under the following conditions.
[Condition]
1) Program setting in PLC parameter dialog box
   1: ABC (sequence) <scan>
   2: DEF (SFC) <scan>
   3: XYZ (sequence) <low speed>
2) Low speed program time setting in parameter: 5ms
3) SFC program: Set to Auto START ON

[Program execution]



REMARKS

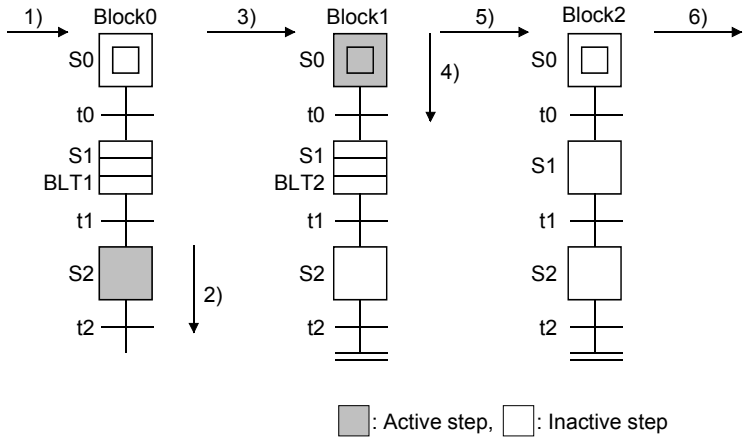
- ∗1: The number of scannable SFC programs in the Universal model QCPU is only one (one normal SFC program). For the Universal model QCPU and LCPU, only one SFC program (one normal SFC program) can be scanned.
- Refer to Section 6.1 for the SFC program start/stop method.

5.3.2 Block execution sequence

(1) In the SFC program, the step in the active block is executed every scan.

(2) When there are multiple blocks, the blocks are processed in order of lower to higher block numbers.
(a) In the active block, the active step in that block is executed.

(b) The inactive block is checked for a START request, and if there is a START request, the block is activated and the step in that block is executed.

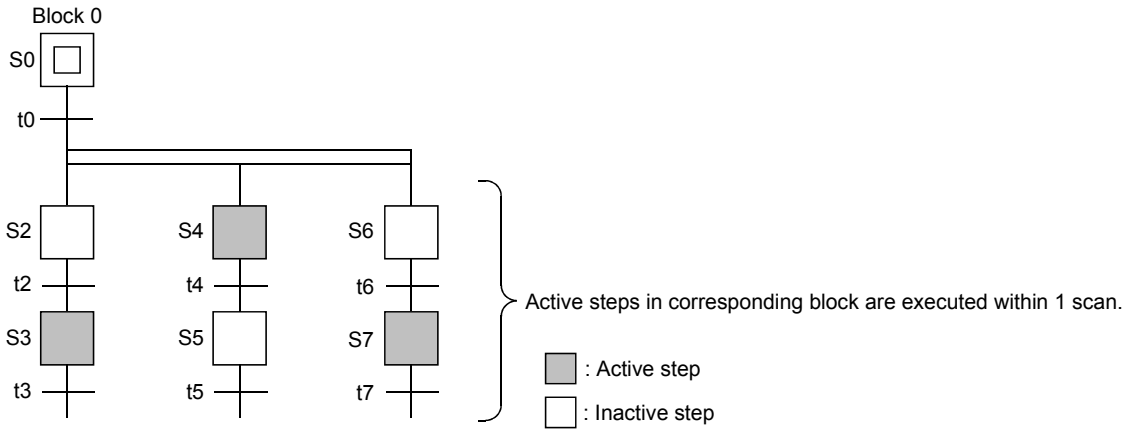


The SFC program is executed in order of 1) to 6).
1): Whether block 0 is active or inactive is checked.
2): Since block 0 is active, the active step (S2) is executed.
3): Whether block 1 is active or inactive is checked.
4): Since block 1 is active, the active step (S0) is executed.
5): Whether block 2 is active or inactive is checked.
6): Since block 2 is inactive, whether the next block is active or inactive is checked.

5 - 10

5.3.3 Step execution sequence

(1) In the SFC program, the operation outputs of all active steps are processed within one scan.



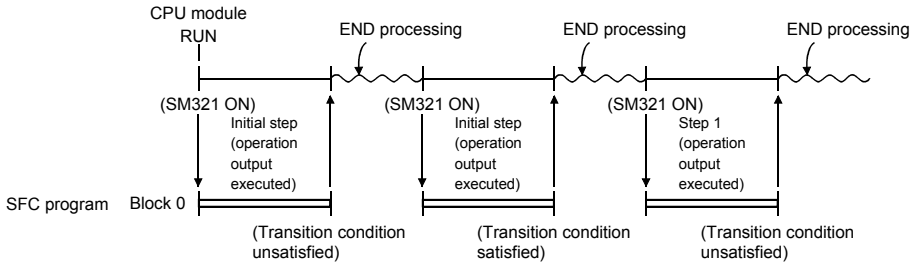
(2) At the end of the operation output execution at each step, whether the transition condition to the next step is satisfied or not is checked.

(a) When the transition condition is not yet satisfied, the operation output of the same step is also executed in the next scan.

(b) When the transition condition is satisfied, the outputs turned ON by the OUT instruction at the executed steps are all turned OFF.
When the next scan is executed, the operation output of the next step is executed.
At this time, the operation output of the step executed previously is deactivated (unexecuted).

The CPU module processes only the program of the operation output of the currently active step and the transition condition to the next step.

Example: The execution sequence from a program start till a transition from the initial step to step 1 is as shown below.



REMARKS

- The step whose attribute has been set to a HOLD step is not deactivated (unexecuted). Processing continues according to the set attribute.

5 - 11

5.3.4 Continuous transition ON/OFF operation

There are two types of SFC program transition processing: "with continuous transition" and "without continuous transition".
Set "with continuous transition" or "without continuous transition" using the continuous transition bit of the SFC information devices.
When the device set to the continuous transition bit is turned ON/OFF by the user, operation is performed as described below.

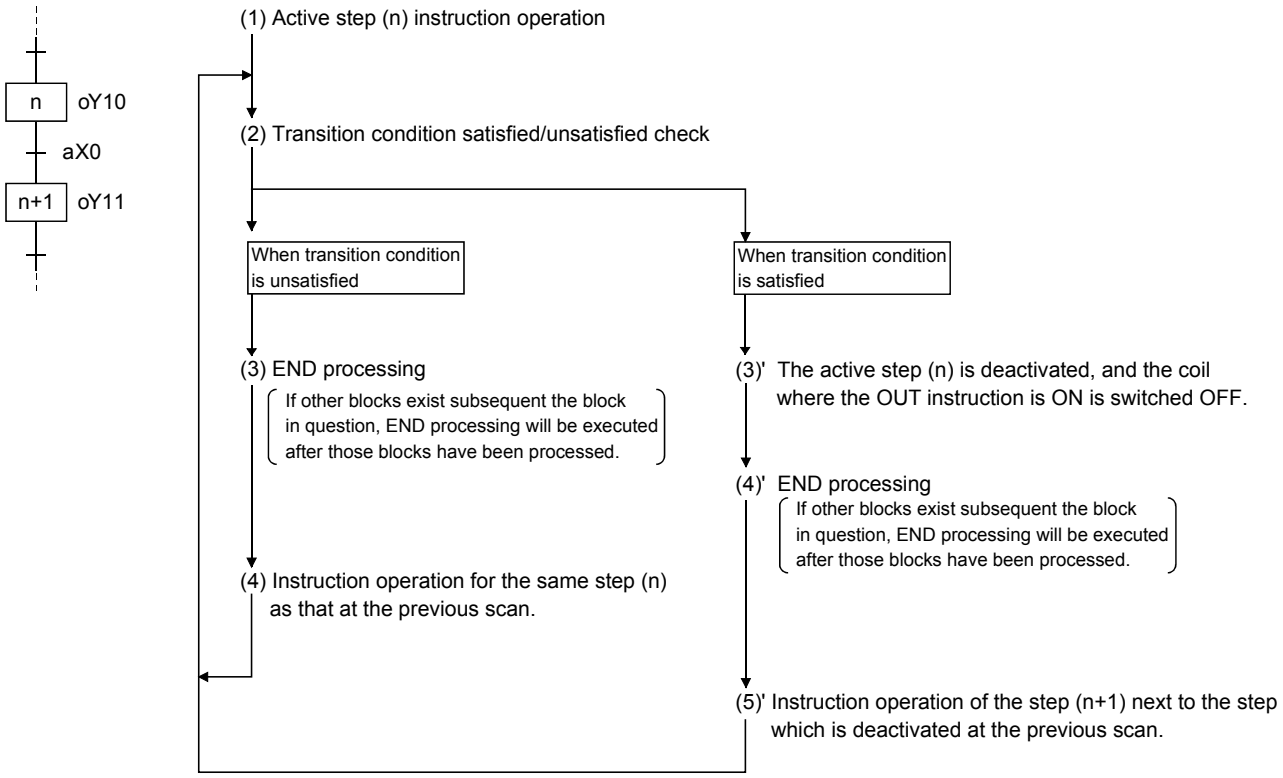
Continuous Transition Bit	SM323	Operation	
No setting	OFF	Without continuous transition	When the transition condition is satisfied, the operation output of the transition destination step is executed in the next scan.
	ON	With continuous transition	When the transition condition is satisfied, the operation output of the transition destination step is executed within the same scan. When the transition conditions of the steps are satisfied continuously, the operation outputs are executed within the same scan until the transition condition is not satisfied or the end step is reached.
OFF	ON/OFF	Without continuous transition	When the transition condition is satisfied, the operation output of the transition destination step is executed in the next scan.
ON	ON/OFF	With continuous transition	When the transition condition is satisfied, the operation output of the transition destination step is executed within the same scan. When the transition conditions of the steps are satisfied continuously, the operation outputs are executed within the same scan until the transition condition is not satisfied or the end step is reached.

POINT

The tact time can be shortened by setting "with continuous transition".
This resolves the problem of waiting time from when the transition condition is satisfied until the operation output of the transition destination step is executed.
However, when "with continuous transition" is set, the operations of the other blocks and sequence program may become slower.
Refer to Section 4.5.5 for details of continuous transition.

(1) Transition processing for continuous transition OFF setting

The SFC program processing procedure without continuous transition will be explained.



POINT

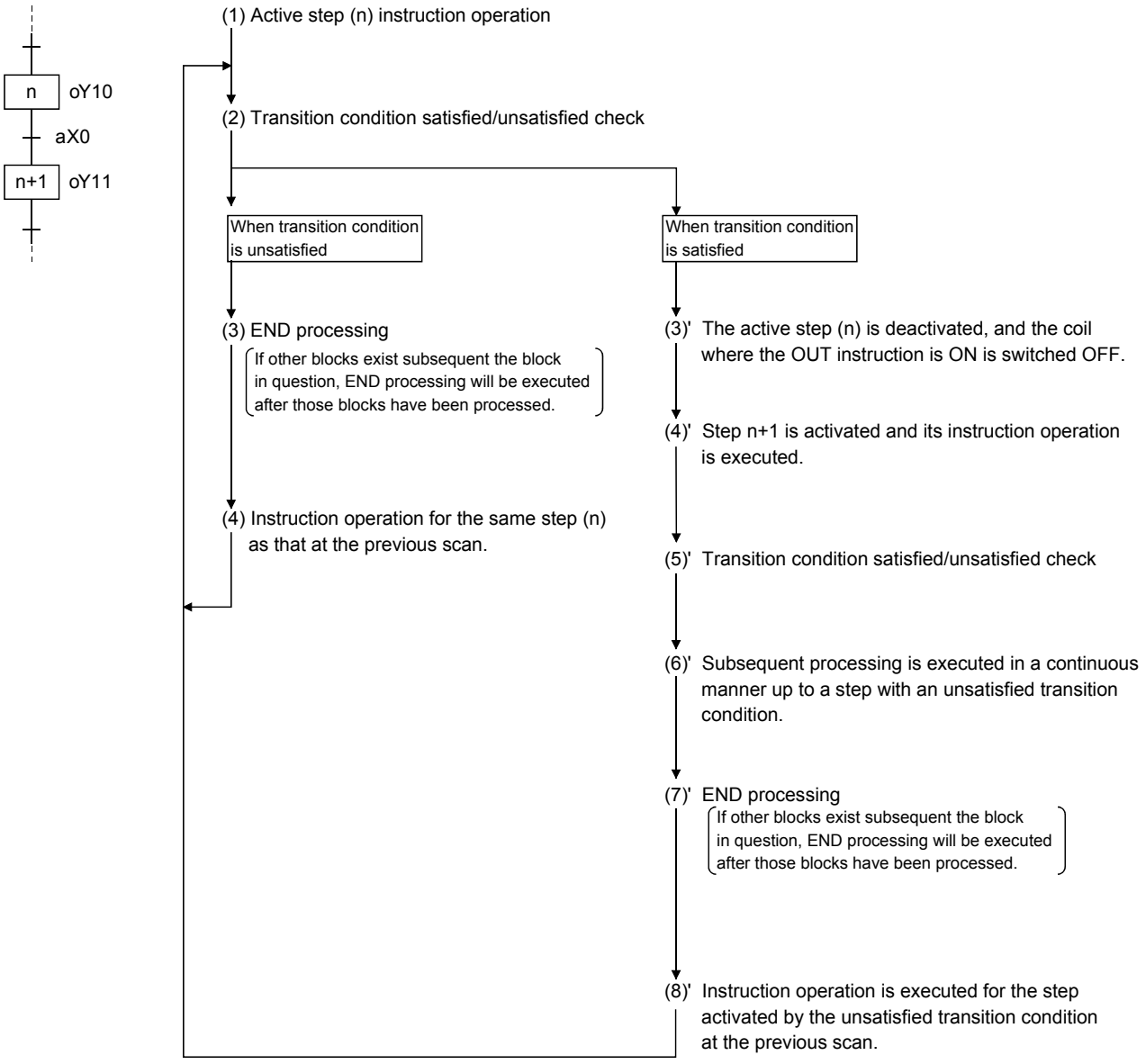
END processing is performed after all the program files set to the "scan execution type" in the program setting of the PLC parameter dialog box have been executed.

Refer to the QCPU User's Manual (Function Explanation, Programming Fundamentals) for the detailed processing order of the programs other than the SFC program and their processings

5 - 13

(2) Transition processing for “continuous transition ON” setting

The SFC program processing procedure with continuous transition will be explained.



POINT

END processing is performed after all the program files set to the "scan execution type" in the program setting of the PLC parameter dialog box have been executed.
Refer to the QCPU User's Manual (Function Explanation, Programming Fundamentals) for the detailed processing order of the programs other than the SFC program and their processings.

5 - 14

6. SFC PROGRAM EXECUTION

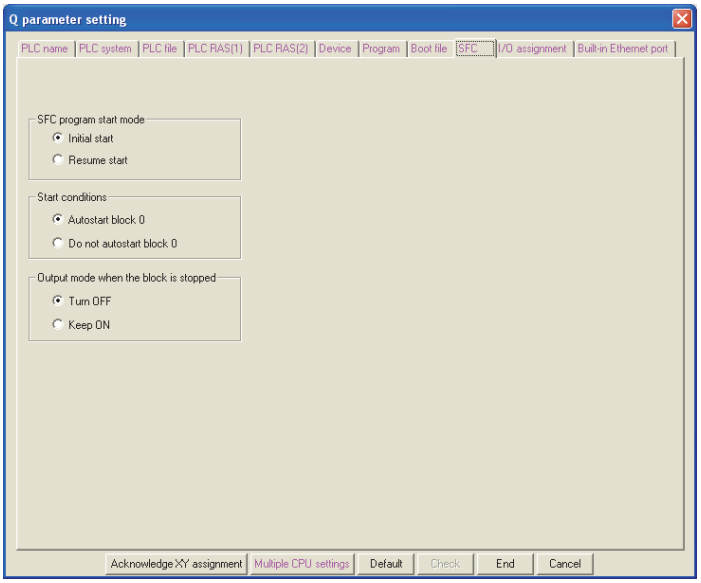
6.1 SFC Program START and STOP

There are the following four types of SFC program start and stop methods.
- Auto START using PLC parameter
- Start and stop using the special relay for SFC program start/stop (SM321)
- Start and stop using the PSCAN/POFF instruction (except the Basic model QCPU)
- Start and stop using the programming tool (except the Basic model QCPU, Universal model QCPU, and LCPU)

(1) Auto START using PLC parameter

Set the start condition in the "SFC setting" of the PLC parameter dialog box to "Block 0 Auto START".

The SFC program is started when the CPU module switches from STOP to RUN.
(When the SFC program starts, block 0 also starts.)



(2) Start and stop using the special relay for SFC program start/stop (SM321)

SM321 turns ON when an Auto START is made using the PLC parameter.

(a) Turn OFF SM321 to stop the SFC program execution.

(b) Turn ON SM321 to start the SFC program.

(3) Start and stop using the PSCAN/POFF instruction (except the Basic model QCPU)

SM321 turns ON when an Auto START is made using the PLC parameter.

(a) When the POFF instruction is executed, the SFC program in execution turns off the output and then stops.
The execution type changes to the "stand-by type".

(b) When the PSCAN instruction is executed, the stand-by type SFC program can be started.
However, when the SFC program has not been set to the "scan execution type" (SM321 is OFF) in the program setting of the PLC parameter dialog box, the SFC program is started by turning ON Sm321.
The execution type changes to the "scan execution type".

6.1.1 SFC program resumptive START procedure

The SFC program START format can be designated as “initial START” or “resumptive START”. The “resumptive START” setting procedure as well as some precautions regarding the “resumptive START” format are described below.

(1) Resumptive START setting procedure
Make the resume START setting of the SFC program in the "SFC program start mode" of the SFC setting in the PLC parameter dialog box.

(2) Block operation status resulting from “SFC program START mode” setting
At an SFC program start, whether an initial start or resume start will be made is determined by the combination of the setting of the "SFC program start mode" in the PLC parameter dialog box and the ON/OFF status of the "special relay for setting SFC program start status (SM322)".

SFC Program Start Mode / Operation	Initial Start		Resume Start	
	SM322: OFF (Initial status) ∗1	SM322: ON (When changed by user)	SM322: ON (Initial status) ∗1	SM322: OFF (When changed by user)
SM321: Turned ON	Initial	Initial	Resume	Initial
Programmable controller: Powered ON			Resume/Initial ∗3	Initial
Programmable controller: Powered OFF and then ON after SM321 is switched from ON to OFF or the CPU module is switched from RUN to STOP			Resume ∗2	Initial
CPU module: Reset and RUN			Resume/Initial ∗6	Initial
CPU module: Reset and RUN after SM321 is switched from ON to OFF or the CPU module is switched from RUN to STOP			Resume ∗2	Initial
CPU module: Switched from STOP to RUN	Resume			
CPU module: STOP, write a program, and then RUN	Initial			

Initial: Initial start, Resume: Resume start

∗1: SM322 is turned ON/OFF by the system according to the setting of the "SFC program start mode" in the PLC parameter dialog box when the CPU module switches from STOP → RUN.
- At initial start setting: OFF
- At resume start setting: ON

∗2: Operation at resume start
The held coil HOLD step SC becomes inactive, and is not kept held.
In the Basic model QCPU, Universal model QCPU and LCPU, the held coil HOLD step SC restarts in the held status.
However, the output is not held. To hold the output, make latch setting for the devices desired to be held.
At a resume start, the SFC program stop position is held but the status of each device used for the operation output is not held.
Therefore, make latch setting for the devices whose statuses must be held in making a resume start.

∗3: Depending on the timing, a resume start is disabled and an initial start may be made.
To perform a resume start, turn ON and then OFF SM321 or switch the CPU module from RUN to STOP, and power OFF and then ON the programmable controller.
An initial start is always performed in the Basic model QCPU and the Universal model QCPU with serial number (first five digits) "11042" or earlier.

6 - 2

∗4: A resume start may be made depending on the SFC program change.
If a resume start is made as-is, a start is made from the old step number, leading to a malfunction of the mechanical system.
When any SFC program change (SFC diagram correction such as step addition and deletion) has been made, make an initial start once and then return it to a resume start.
An initial start is always performed in the Basic model QCPU and the Universal model QCPU with serial number (first five digits) "11042" or earlier.

∗5: In the Universal model QCPU and LCPU, a resume start is performed if data other than SFC programs are changed.

∗6: The Basic model QCPU and Universal model QCPU of which the first 5 digits of the serial number are "11042" always makes an initial start.

POINTS

(1) When the programmable controller is powered OFF or the CPU module is reset, the intelligent function modules and special function modules are initialized.
When making a resume start, create an initial program for the intelligent function module/special function module in the block that is always active or in the sequence program.

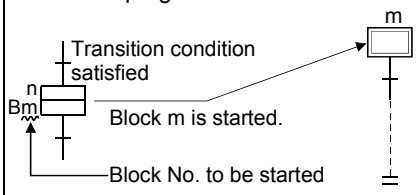
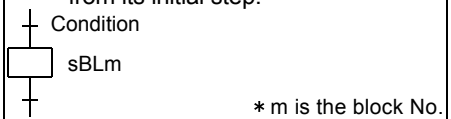
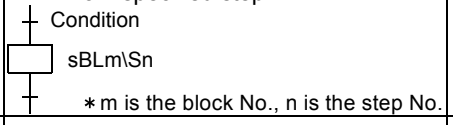
(2) When the programmable controller is powered OFF or the CPU module is reset, values in the devices without a latch setting are cleared.
To hold the values in the SFC information devices, set a latch range.

6.2 Block START and END

6.2.1 Block START methods

The block START methods during SFC program execution are described below.

As shown below, there are several block START methods. Choose the method which is most suitable for the purpose at hand.

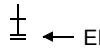
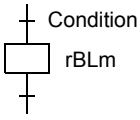
START Method	Operation Description	Remarks	Block 0	Other than Block 0
Auto START using PLC parameter	• By setting the "start condition" to "block 0 Auto START" in the SFC setting of the PLC parameter dialog box, block 0 is automatically started at an SFC program start, and processing is executed from the initial step.	• Convenient when block 0 is used as a control block, a preprocessing block, or a constant monitoring block, for example.	○	×
Block START by SFC diagram symbol	• Another block is started by the block START steps (Bm, Bm) at each of the SFC program blocks. 	• Convenient when the sequence control is clear as in automatic operation. • There are 2 types of block START: The START source step remains active until the START destination block is ended. The START source transition occurs without waiting for the START destination block to be ended (SFC diagram symbol: Bm).	○	○
Block START by SFC control instruction	• Using an SFC control instruction, a specified block is forcibly started from an SFC program step (operation output), or from another sequence program. (1) When specified block is executed from its initial step:  (2) When specified block is executed from specified step: 	• Convenient when starting an error reset processing block at error detection, etc., and for executing interrupt processing, for example.	○	○
Block START by SFC information device	• The corresponding block is activated by forcibly turning ON the "block START/END bit", which was set to each block as the SFC information device, in the program or peripheral device.	• Convenient for debugging and test operations in 1-block units because the block can be started from a peripheral device without requiring a program.	○	○

○: Usable, ×: Unusable

6.2.2 Block END methods

The methods for ending block operations are described below.

As shown below, there are several block END methods. Choose the method which is most suitable for the purpose at hand.

END Method	Operation Description	Remarks
Block END by SFC diagram symbol	• Block processing is ended and the block is deactivated when the block's END step is executed. 	• Convenient for cycle stops at automatic operations, etc. • Multiple END steps are possible within a single block.
Block END by SFC control instruction	• Using an SFC control instruction, a specified block is forcibly ended and deactivated from an SFC program step (operation output), or from another sequence program.  ∗ m is the block No. ∗ Block processing is also ended when the rBLm□Sn instruction is used to deactivate all the active steps of the corresponding block.	• Convenient for executing a forced STOP (at emergency stops, etc.) without regard to the operation status.
Block END by SFC information device	• The processing of the corresponding block is ended to deactivate it by forcibly turning OFF the "block START/END bit", which was set to each block as the SFC information device, in the program or peripheral device.	• Convenient for debugging and test operations because block processing can be ended from a peripheral device without requiring a program.

POINTS

(1) A forced end to block processing is possible using a method which is different from that used to start the block.
Example: 1) A block started by an SFC diagram symbol (Bm⊟, Bm⊟) can be ended by an SFC control instruction (nBLm).
2) A block started by an SFC control instruction (sBLm) can be ended by forcibly turning OFF the block START/END bit of the SFC information devices.

(2) After block END processing is completed, the block can be restarted as shown below.

Block		
Block 0	When the Start conditions is designated as “Autostart block 0”	• After block processing is ended, processing is started automatically from the initial step.
	When the Start conditions is designated as “Do not autostart block 0”	• After block processing is ended, the block remains inactive until a START request occurs by one of the methods described in Section 6.2.1.
Other than block 0		

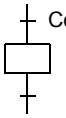
6.3 Block Temporary Stop and Restart Methods

6.3.1 Block STOP methods

The temporary block STOP methods which can be used during SFC program execution are described below.

(1) Block STOP methods

The methods for temporarily stopping a block during SFC program operation are shown below.

STOP Method	Operation Description	Remarks
Block STOP by SFC control instruction	• Using an SFC control instruction, a specified block is temporarily stopped from an SFC program step (operation output), or from another sequence program.  ∗ m is the block No.	• Convenient for temporarily stopping operation (at error detection, etc.) in order to correct the error by manual operation. The manual operation control program can be placed at another block which is forcibly started when the block STOP occurs.
STOP by SFC information device	• The execution of the specified block is temporarily stopped by forcibly turning ON the "block STOP/RESTART bit", which was set to each block as the SFC information device, in the program or peripheral device.	• Convenient for confirming operation by step control at debugging and test operations, because block processing can be stopped from a peripheral device without requiring a program.

6 - 6

(2) Block STOP timing and coil output status when STOP occurs

The STOP timing in response to a block STOP request, and the coil output status during the STOP are as shown below.

<table>
<tr><th rowspan="3">Setting of Output Mode at Block Stop in PLC Parameter</th><th rowspan="3">Operation Output at Block Stop (SM325)</th><th rowspan="3">Status of STOP-time Mode Bit</th><th colspan="3">Operation</th></tr>
<tr><th>Active step other than held step
(including HOLD step whose transition condition is not satisfied)</th><th colspan="2">Held step ＊</th></tr>
<tr><th>Coil HOLD step (SC)</th><th>Operation HOLD step (without transition check) (SE)</th><th>Operation HOLD step (with transition check) (ST)</th></tr>
<tr><td rowspan="2">• Turns OFF (coil output OFF)
• Remains ON (coil output held)</td><td rowspan="2">• OFF (coil output OFF)</td><td>• OFF
• No setting (immediate stop)</td><td>• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status remains active.</td><td rowspan="2">• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status becomes inactive.</td><td colspan="2" rowspan="2">• Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped.
• The status remains active.</td></tr>
<tr><td>• ON (STOP after transition)</td><td>• Normal operation is performed until the transition condition is satisfied.
• When the transition condition is satisfied, the end processing of the corresponding step is performed.
At the same time, the transition destination step becomes active and the block stops immediately.</td></tr>
<tr><td rowspan="2">• Remains ON (coil output held)</td><td rowspan="2">• ON (coil output held)</td><td>• OFF
• No setting (immediate stop)</td><td>• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held.
• The status remains active.</td><td colspan="3" rowspan="2">• Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held.
• The status remains active.</td></tr>
<tr><td>• ON (STOP after transition)</td><td>• Normal operation is performed until the transition condition is satisfied.
• When the transition condition is satisfied, the end processing of the corresponding step is performed.
At the same time, the transition destination step becomes active and the block stops immediately.</td></tr>
</table>

＊: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

POINT

The operation of SM325 differs depending on the CPU module.

- For the Basic model QCPU, High Performance model QCPU, and Process CPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) at STOP RUN of the CPU module.
- For the Universal model QCPU and LCPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) when the CPU module is powered ON or is reset.

Parameter setting	SM325
Turns OFF (coil output OFF)	OFF
Remains ON (coil output held)	ON

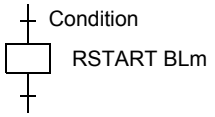
Note that the output mode at block stop can be changed regardless of the parameter setting by turning ON/OFF SM325 in the user program.

6.3.2 Restarting a stopped block

The methods for restarting a block which has been temporarily stopped during SFC program processing are described below.

(1) Restarting block processing

The methods for restarting a block which has been temporarily stopped are shown below.

Restart Method	Operation Description	Remarks
Restart by SFC control instruction	• Processing of the specified block is restarted by an SFC control instruction at a step (operation output) or sequence program outside the stopped block.  	• Convenient for returning to automatic operation when the manual control END signal is output at the temporary STOP.
RESTART by SFC information device	• The execution of the corresponding block is restarted by forcibly turning OFF the "block STOP/RESTART bit", which was set to each block as the SFC information device, in the program or peripheral device.	• Convenient for confirming operation by step control at debugging and test operations, because block processing can be restarted from a peripheral device without requiring a program.

(2) Active step when restart occurs

The step which is active when a block is restarted varies according to the status which existed when the STOP occurred, as shown below.

Output Mode Setting at Block STOP	Operation Output at Block RESTART			
	Active step other than held step (including HOLD step whose transition condition is not satisfied)	Held step ∗		
		Coil HOLD step (SC)	Operation HOLD step (without transition check) (SE)	Operation HOLD step (with transition check) (ST)
At coil output OFF	• Returns to normal operation.	• Restart disabled. (Since the step is deactivated at a block STOP)	• Restarts the execution of the operation output in a HOLD status.	• Restarts the operation output in a HOLD status. • Also checks the transition condition.
At coil output HOLD		• Restarts as held.		

∗: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

6 - 9

POINT

The operation of SM325 differs depending on the CPU module.

- For the Basic model QCPU, High Performance model QCPU, and Process CPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) at STOP RUN of the CPU module.
- For the Universal model QCPU and LCPU
  SM325 turns ON/OFF according to the parameter setting (output mode setting at block stop) when the CPU module is powered ON or is reset.

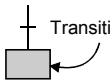
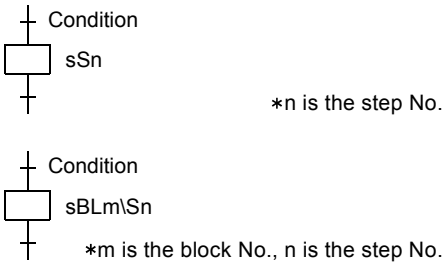
Parameter setting	SM325
Turns OFF (coil output OFF)	OFF
Remains ON (coil output held)	ON

Note that the output mode at block stop can be changed regardless of the parameter setting by turning ON/OFF SM325 in the user program.

6.4 Step START (Activate) and END (Deactivate) Methods

6.4.1 Step START (activate) methods

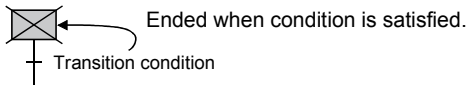
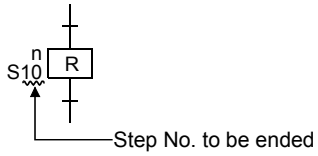
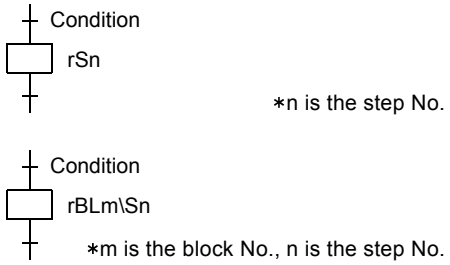
There are the following step START (activation) methods.

Step START (Activation) Method	Operation	Remarks
Step START by SFC diagram symbol	• The corresponding step is automatically started when the preceding transition condition is satisfied.  	• Basic operation of SFC program
Step START by SFC control instruction	• The specified step is forcibly started by the SFC control instruction at the step (operation output) of the SFC program or in another sequence program.  	• Jump to other blocks can be made. • When the block of the destination step is inactive, a block forced START is made from the specified step. • When there are initial steps in multiple blocks, a selection START is made.

6 - 11

6.4.2 Step END (deactivate) methods

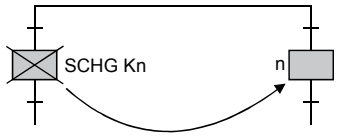
Steps can be ended (deactivated) by the methods shown below.

END Method	Operation	Remarks
END by SFC diagram symbol	• The step is automatically ended by the system when the transition condition associated with the corresponding step is satisfied. 	• Basic operation of SFC program • When the step attribute has been specified, operation is performed according to the attribute.
	• Set the step to a reset step as the step attribute and specify the step number to be ended. 	• Convenient for ending the HOLD step when the machine operation condition is satisfied during SFC program execution, when a transition to the error processing step is performed by selection branch, for example. • The step number to be ended can be specified in only the same block.
END by SFC control instruction	• The specified step is forcibly ended by the SFC control instruction at the step (operation output) of the SFC program or in another sequence program. 	• The steps in different blocks can also be ended. • The block is ended when all steps of the corresponding block are deactivated by the RST instruction.

6 - 12

6.4.3 Changing an active step status (Not available for Basic model QCPU, Universal model QCPU, and LCPU)

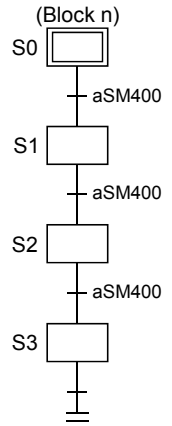
This section explains the method for ending (deactivating) an active step and starting (activating) the specified step.

Changing Method	Operation	Remarks
Change by SFC control instruction	• At the step (operation output) of the SFC program, the instruction execution step is ended and the specified step is forcibly started.  Instruction execution step is deactivated. ⇒ Specified step is activated.	• Convenient when the jump destination changes depending on the condition. • The change destination step can be specified within the current block. • Indirect designation (D0, K4M0, etc.) can also be used to specify the change destination step. • When multiple instructions have been described within one step, the change destination executed in the same can will be valid.

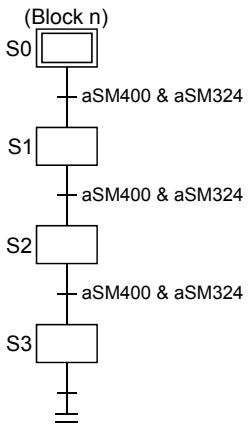
6.5 Operation Methods for Continuous Transition

If "with continuous transition" is set, whether a continuous transition will be performed or not can be selected at each step using the continuous transition disable flag (SM324).

(1) Processing performed when continuous transition disable flag is not used

SFC Program	With Continuous Transition	Without Continuous Transition
	When the corresponding block becomes active, the processings of all steps are executed in the same scan, and end step processing is performed to deactivate the block.	• When the corresponding block becomes active, steps are executed in a 1-step-per-scan format. • The end step processing is performed in the third scan to deactivate the block.

(2) Processing performed when continuous transition disable flag is used

SFC Program	With Continuous Transition	Without Continuous Transition
	• When the corresponding block becomes active, execution proceeds to step 1 since SM324 is ON. When execution proceeds to step 1, the processing of the first scan is ended since SM324 turns OFF. • In the second scan, execution proceeds to step 2 since SM324 turns ON again. When execution proceeds to step 2, SM324 turns OFF. Since the transition condition of step 2 does not have the contact of SM324, a transition occurs and the end step processing is performed to deactivate the block.	• When the corresponding block becomes active, steps are executed in a 1-step-per-scan format independently of whether SM324 is present or absent. • The end step processing is performed in the third scan to deactivate the block.

6 - 14

6.6 Operation at Program Change

The SFC program of the CPU module can be changed in either of the following methods.
• Write to PLC (write in file unit)
• Online change (write in ladder block unit)
• Online change (inactive block) ∗1

The following table indicates SFC program changes that can be made in the above methods.

○: Possible, ×: Impossible

Change Type \ Function			Program Change by Write to PLC		Program Change by Online Change	Online change (inactive block)
			PAUSE/STOP status	RUN status ∗2 ∗3		
SFC program addition			○	×	×	×
SFC block addition/deletion			○	○	×	○
SFC block change	SFC diagram change	Step/transition addition/deletion	○	○	×	○∗4
		Transition destination change	○	○	×	○∗4
		Step attribute change	○	○	×	○∗4
	Change in SFC diagram	Operation output sequence program change	○	○	○	○
		Transition condition sequence program change	○	○	○	○
	Block data change		○	○	×	○

○: Possible, ×: Impossible

6 - 15

POINT

∗1: The LCPU does not support this function.

∗2: This function can be executed only when a CPU module and programming tool are used in the following combination.

CPU module	Programming tool
High Performance model QCPU (whose first five digits of serial No. are 04122 or later)	GX Developer Version 8 or later GX Works2
Process CPU (whose first five digits of serial No. are 07032 or later)	GX Developer Version 8 or later
Redundant CPU	Version 8.18U or later

∗3: The Universal model QCPU and LCPU do not support the use of this function in the RUN status.

∗4: This function can be executed only when a CPU module and GX Works2 are used in the following combination.

CPU module	Programming tool
Universal model QCPU other than the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU (serial number (first five digits) is "12052" or later)	GX Works2 Version 1.34L or later

6 - 16

6.6.1 Operation at program change made by write to PLC

(1) When program was written with CPU module in PAUSE/STOP status

(a) Program start after write to PLC

An initial start is performed independently of the SFC start mode setting (initial start/resume start).

Depending on the SFC program change, however, an initial start is not made but a resume start may be made at the resume start setting.

Refer to Section 4.7.1 for details of the SFC program start mode.

(b) Device status at program start

At a program start after write to PLC, the CPU module devices operate as described in the following table depending on the setting of the SFC device clear mode setting flag (SM326).

SM326	Operation	
	Step relay	Other than step relay
OFF	Turned ON/OFF by the system.	SFC program is executed after all devices have been cleared.
ON	Turned ON/OFF by the system.	SFC program is executed with all devices held.

POINT

The setting of SM326 is valid only when an SFC program exists after write to PLC.
When sequence program and/or parameter write is performed, the setting of SM326 is also valid.
(The setting of SM326 is ignored when only the data other than the SFC program, sequence program and parameters are written.)

(2) When program was written with CPU module in RUN status

(a) Program start after write to PLC

An initial start is performed independently of the SFC start mode setting (initial start/resume start).

Refer to Section 4.7.1 for details of the SFC program start mode.

(b) Device status at program start

The SFC program is executed with all devices held.

6.6.2 Program change by online change

(1) Program start after write to PLC

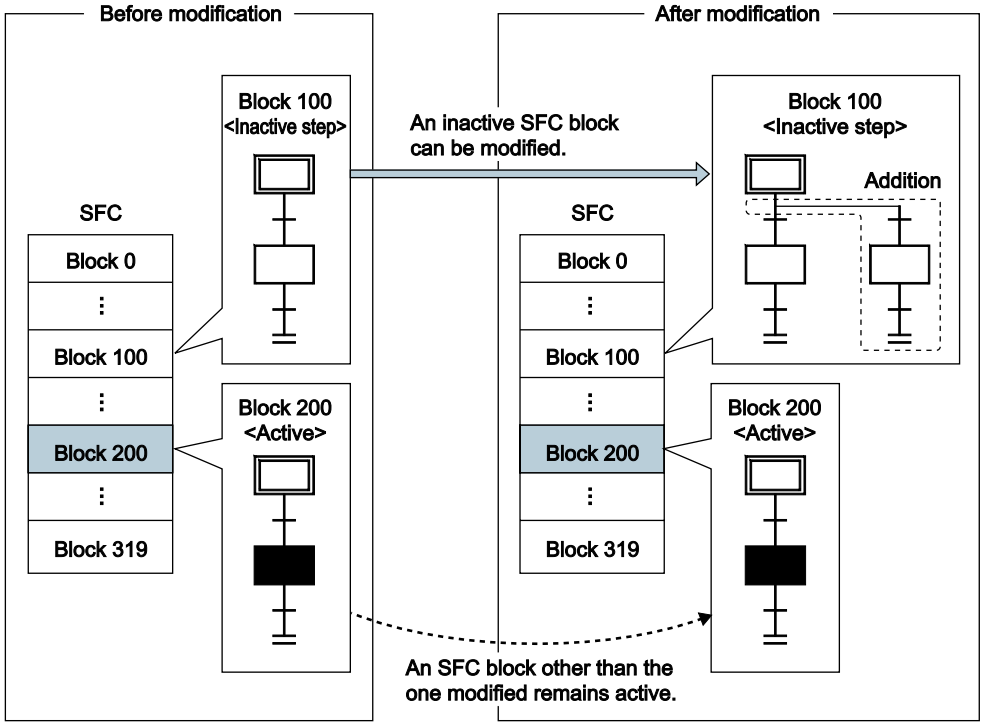
When program change is made by online change, a resume start is performed independently of the SFC start mode setting.

(2) Device status at program start

The SFC program is executed with all devices held.

6.6.3 Online change (inactive block)

An inactive SFC block can be changed in units of blocks.



(1) Supported program

This function can be executed to an SFC program registered in the Program tab of the PLC parameter dialog box.

POINT

When there are multiple programs in the program memory, executing this function to a program not registered in the Program tab will result in a communication error.

(2) Available operations

The following operations can be executed to an inactive block with GX Works2.

Operation	Description
Changing a block	• An SFC block program in the CPU module can be changed. • An SFC information device for the target SFC block can be changed.
Adding a block	• An SFC block can be added to an SFC program in the CPU module. • An SFC information device can be added to the target SFC block.
Deleting a block	• The specified SFC block can be deleted from the SFC program in the CPU module. • An SFC information device for the target SFC block can be deleted. • When the target block is not in an SFC program in the CPU module, a communication error will occur.

6 - 19

(3) Area to be overwritten

(a) Area to be changed

All programs of the target block are overwritten.

Multiple blocks cannot be batch-written.

In online change (inactive block), a program (before change) in a programming tool is not verified with the program in the CPU module.

Therefore, verifying an SFC program in the programming tool with that in the CPU module beforehand is recommended.

(b) Change in signal flow memory

Signal flow memories of the target block all turn off.

(c) SFC information devices

The following SFC information devices can be added/changed/deleted.

1) Block START/END bit
2) Step transition bit
3) Block PAUSE/RESTART bit
4) Pause mode bit
5) Number of active steps register
6) Continuous transition bit

POINT
Before an SFC program is changed, devices specified in 1) to 6) are checked if they are within the device range. If any of them are outside the device range, a communication error will occur.

(4) Available execution type

Online change (inactive block) can be executed to a scan execution type program (cannot be executed to a standby type program).

(5) Changing the execution type of a program during online change (inactive block)

The execution type of a program being written by online change (inactive block) cannot be changed with Program control instructions (POFF and PSCAN instructions).

(6) Availability depending on block status

The following table shows availability of online change (inactive block) depending on the block status at the start of writing.

Block status	Availability
Inactive[*1]	Online change (inactive block) can be executed.
Active	Online change (inactive block) cannot be executed. However, while SM321 (Start/stop SFC program) is off, even when the block is active immediately before the relay turns off, online change (inactive block) can be executed.[*2]

∗1: For how to end processing of a block and set it to inactive, refer to Section 6.2.2.

∗2: While SM321 is off, online change (inactive block) can be executed, regardless of the target block status immediately before the relay turns off. Note when online change (inactive step) is executed while SM321 is off, the SFC program always starts in initial start mode, regardless of settings configured in "SFC Program Start Mode" in the SFC tab of the PLC parameter dialog box and SM322 (SFC program start status).

(7) Operation when the target block is attempted to be started while online change (inactive block) is executed

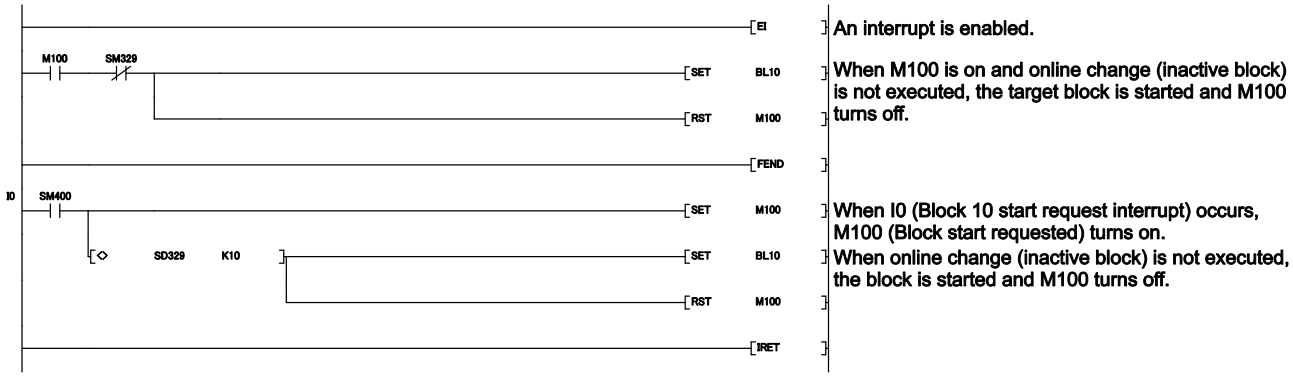
The target block does not start. The following table shows operations depending on block start method.

Start method (activation method)	Operation at block start
Block START step (without END check)	• The target block does not start until when online change (inactive block) is ended. Even when the transition condition for the step is met, the processing does not move to the next step. • The target block starts after online change (inactive block) is ended. When the transition condition is met, the processing moves to the next step.
Block START step (with END check)	• The target block does not start until when online change (inactive block) is ended. • The target block starts after online change (inactive block) is ended. When the transition condition for the step is met, the processing moves to the next step.
SFC control instruction (SET BL □, SET S □, SET BL □ \S □ instructions)	The target block does not start. While an instruction contact remains on, the target block starts after online change (inactive block) is ended (for a program to execute the Block START instruction, refer to the program example in Section 6.6.3(7)(a).
SFC information device (block START/END bit)	The target block does not start even when the block START/END bit turns on. When the block START/END bit is on, the target block starts after online change (inactive block) is ended. (The status of the block START/END bit does not change until when online change (inactive block) is ended).

POINT

In the STOP or PAUSE status, an active step holds the activated status. Therefore, when the CPU module is set to STOP or PAUSE while the target block is active, online change (inactive block) cannot be executed to the block.

(a) Program example to execute the Block START instruction during online change (inactive block)



6 - 22

(8) Reserved area for online change

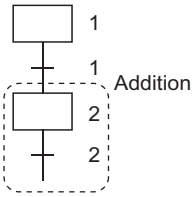
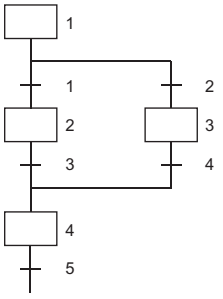
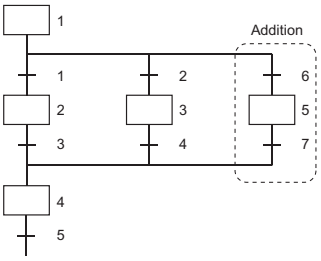
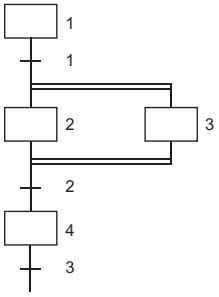
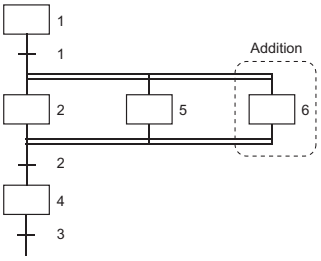
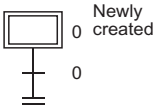
Secure reserved area for online change by the amount to be added/changed by online change (inactive block).

(a) Adding/changing an SFC information device

When all SFC information devices are not set for the target block, SFC information device area will not be created in the program file. The device area will be added to the program file when an SFC information device is added to the block by online change (inactive block). Then, free area in reserved area for online change will be reduced. (When the target block has already contained SFC information devices, changing the SFC information device will not reduce free area in reserved area for online change.)

(b) Amount used in reserved area for online change by adding an SFC diagram symbol

The following table shows the amount used in reserved area for online change by adding an SFC diagram symbol by online change (inactive block).

Description	Before addition	After addition	Amount used in reserved area for online change (unit: step)
Adding an SFC step in serial transition			7 + Number of steps in step 2 + Number of steps of transition condition 2
Increasing the number branches in selection branching			12 + Number of steps in step 5 + Number of steps of transition condition 6 + Number of steps of transition condition 7
Increasing the number of branches in parallel branching			7 + Number of steps in step 6
Adding an SFC block	(No relevant SFC blocks)		10 + Number of steps in step 0 + Number of steps of transition condition 0 (When SFC information device is set, further 9 steps are reduced.)

6 - 23

(9) Precautions

(a) Online change from another GX Works2

If GX Works executes online change (inactive block) while another GX Works2 executes online change or program backup, a communication error will occur. The same occurs if another GX works executes online change or program backup while GX Works2 executes online change (inactive block).

(b) Excess of the number of SFC steps

After addition of SFC steps, if the number of SFC steps exceeds the number of step relay (S) points set in the Device tab of the PLC parameter dialog box, a communication error will occur.

(c) Online change (inactive block) during boot operation

When online change (inactive block) is executed during boot operation from a memory card, the original program in the memory card will not be changed.

(d) Operation if the CPU module is powered off and then on or is reset before termination of online change (inactive block)

The changes in the target program will not be reflected.

(e) Online change (inactive block) immediately after writing to the CPU module

When the CPU module is set to STOP and an SFC program is written to the CPU module, online change (inactive block) cannot be executed to the SFC program until when the CPU module is set to RUN.

If executed, a communication error will occur.

(f) Online change (inactive block) while "CAN'T EXE.PRG." (error code: 2504) occurs

When multiple SFC programs of scan execution type are registered and "CAN'T EXE.PRG." (error code: 2504) occurs, online change (inactive block) cannot be executed. If executed, a communication error will occur.

MEMO

6 - 25

APPENDICES

APPENDIX 1 Special Relay and Special Register List

This section lists the special relays and special registers that can be used in SFC programs.
For the special relays and special registers for other programs, refer to the user's manual for the CPU module used.

The heading descriptions in the lists are shown in the table below.

Item	Function of Item
Number	• Indicates special relay and special register number.
Name	• Indicates name of special relay and special register.
Meaning	• Indicates contents of special relay and special register.
Explanation	• Discusses contents of special relay and special register in more detail.
Set by (When set)	• Indicates whether the relay or register is set by the system or user, and, if it is set by the system, when setting is performed. <Set by> S : Set by system
	U : Set by user (sequence programs or test operations from GX Developer)
	S/U : Set by both system and user
	<When set> Indicated only for relays and registers set by system Initial : Set only during initial processing (when power supply is turned ON, or when going from STOP to RUN) Status change : Set only when there is a change in status Error : Set when error occurs Instruction execution : Set when instruction is executed
Corresponding CPU	Indicates the corresponding CPU module type name.

A

APP -1

APPENDIX 1.1 Special Relay (SM) List

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU				
					Basic model QCPU	High Performance model QCPU	Process CPU	Redundant CPU	Universal model QCPU, LCPU
SM90	Step transition watch dog timer START (corresponds to SD90)	OFF: Not started (Watch dog timer reset) ON : Started (Watch dog timer start)	Switched ON to begin the step transition watch dog timer count. Watch dog timer is reset when switched OFF.	User	×	○	○	○	×
SM91	Step transition watch dog timer START (corresponds to SD91)								
SM92	Step transition watch dog timer START (corresponds to SD92)								
SM93	Step transition watch dog timer START (corresponds to SD93)								
SM94	Step transition watch dog timer START (corresponds to SD94)								
SM95	Step transition watch dog timer START (corresponds to SD95)								
SM96	Step transition watch dog timer START (corresponds to SD96)								
SM97	Step transition watch dog timer START (corresponds to SD97)								

A

APP -2

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU				
					Basic model QCPU	High Performance model QCPU	Process CPU	Redundant CPU	Universal model QCPU, LCPU
SM98	Step transition watch dog timer START (corresponds to SD98)	OFF: Not started (Watch dog timer reset) ON : Started (Watch dog timer start)	Switched ON to begin the step transition watch dog timer count. Watch dog timer is reset when switched OFF.	User	×	○	○	○	×
SM99	Step transition watch dog timer START (corresponds to SD99)								
SM320	SFC program presence/absence	OFF: Without SFC program ON: With SFC program	• ON if an SFC program has been registered. • OFF if an SFC program has not been registered.	System (initial value)	∗1 ○	○	○	○	○
SM321	SFC program START/STOP	OFF: SFC program not executed (stop) ON: SFC program executed (start)	• The same value as in SM320 is set as the default value. (Automatically switches ON when the SFC program exists.) • When this relay is switched from ON to OFF, the SFC program execution is stopped. • When this relay is switched from OFF to ON, the SFC program execution is restarted.	System (initial value), User					
SM322	SFC program START status	OFF: Initial START ON : Resumptive START	• The SFC program start mode set in the SFC setting of the PLC parameter dialog box is set as the default value. At initial start: OFF At resume start: ON	System (initial value), User					
SM323	All-blocks continuous transition status	OFF: Continuous transition enabled ON : Continuous transition disabled	• Set whether a continuous transition will be performed or not for the block where the "continuous transition bit" of the SFC information devices has not been set	User					
SM324	Continuous transition disable flag	OFF: After transition ON : Before transition	• OFF during operation in the "with continuous transition" mode or during continuous transition, and ON when not during continuous transition. • Always ON during operation in the "without continuous transition" mode.	System (for instruction execution)					

∗1: Available with the CPU module whose function version is B or later

APP -3

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU				
					Basic model QCPU	High Performance model QCPU	Process CPU	Redundant CPU	Universal model QCPU, LCPU
SM325	Operation output at block STOP	OFF: Coil output OFF ON : Coil output ON	Select whether the coil output of the active step will be held or not at a block STOP. • As the default value, OFF when coil output OFF is selected for the output mode at parameter block STOP, and ON when coil output held is selected. • When this relay is OFF, the coil outputs are all turned OFF. • When this relay is ON, the coil outputs are held.	System (initial value), User	○	○	○	○	○
SM326	SFC device clear mode	OFF: Clear device ON : Preserves device	• Select the device status when the CPU is switched from STOP to program write to RUN. (All devices except the step relay)	User					
SM327	Output mode at end step execution	OFF: HOLD step output OFF ON : HOLD step output held	• When this relay is OFF, the SC, SE or ST step that was held when a transition condition had been satisfied turns OFF the coil output when the end step is reached.	User					
SM328	Clear processing mode at arrival at end step	OFF: Clear processing is performed ON : Clear processing is not performed	Select whether clear processing will be performed or not when active steps other than those held exist in the block at the time of arrival at the end step. • When this relay is OFF, the active steps are all ended forcibly to end the block. • When this relay is ON, the execution of the block is continued as is. • When no active steps other than those held exist at the time of arrival at the end step, the held steps are all ended to end the block.	User	○	×	×	×	○

APP -4

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU				
					Basic model QCPU	High Performance model QCPU	Process CPU	Redundant CPU	Universal model QCPU, LCPU
SM331	Normal SFC program execution status	OFF : Not executed ON : Being executed	• Indicates whether the normal SFC program is being executed or not. • Used as an execution interlock of the SFC control instruction.	S (status change)	×	∗2 ○	×	∗4 ○	×
SM332	Program execution management SFC program execution status	OFF : Not executed ON : Being executed	• Indicates whether the program execution management SFC program is being executed or not. • Used as an execution interlock of the SFC control instruction.	S (status change)					
SM735	SFC comment readout instruction in execution flag	OFF : SFC comment readout instruction is inactivated. ON : SFC comment readout instruction is activating.	• Turns on the instructions, (S(P).SFCSCOMR) to read the SFC step comments and (S(P). SFCTCOMR) to read the SFC transition condition comments.	System (status change)	×	∗3 ○	∗4 ○	∗4 ○	×

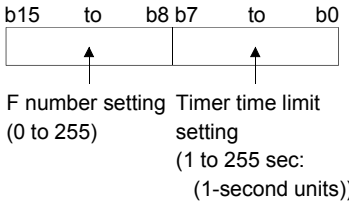
∗1: Available with the CPU module whose function version is B or later

∗2: Available with the CPU module whose serial number (first five digits) is "04122" or later.

∗3: Available with the CPU module whose serial number (first five digits) is "07012" or later.

∗4: Available with the CPU module whose serial number (first five digits) is "07032" or later.

APPENDIX 1.2 Special Register (SD) List

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU				
					Basic model QCPU	High Performance model QCPU	Process CPU	Redundant CPU	Universal model QCPU, LCPU
SD90 SD91 SD92 SD93 SD94 SD95 SD96 SD97 SD98 SD99	Step transition monitoring timer setting value	Timer set value and F No. at time-out	• Set the set time of the step transition watch dog timer and the annunciator No. (F No.) that will turn ON at time-out of the watch dog timer.  • The timer starts when any of SM90 to SM99 is turned ON during an active step, and the set annunciator (F) turns ON if the transition condition following the corresponding step is not satisfied within the timer time limit.	System (at error occurrence)	×	○	○	○	×

The special registers SD90 to SD99 correspond to the following special relays.

Special register	Special relay
SD90	SM90
SD91	SM91
SD92	SM92
SD93	SM93
SD94	SM94
SD95	SM95
SD96	SM96
SD97	SM97
SD98	SM98
SD99	SM99

APPENDIX 2 Restrictions on Basic Model QCPU, Universal Model QCPU, and LCPU and Alternative Methods

This section explains the restrictions on use of SFC programs for the Basic model QCPU, Universal model QCPU, and LCPU.

(1) Function comparison

Item			Basic Mode QCPU, Universal model QCPU, LCPU	High Performance Model QCPU, Process CPU, Redundant CPU	Alternative Method
Step transition watchdog timer			Not provided	Provided	Appendix 2.1
SFC operation mode setting	Operation mode at block double START		Not provided (Fixed to "WAIT")	Provided	–
	Operation mode for transition to active step (at step double START)		Not provided (Fixed to "TRANSFER")	Provided	–
	Periodic execution block setting		Not provided	Provided	Appendix 2.2
SFC control instruction	Forced transition check instruction	aTRn	Not provided	Provided	–
		&aTRn			
		laTRn			
		bTRn			
		&bTRn			
		lbTRn			
		aBLm/TRn			
		&aBLm/TRn			
		laBLm/TRn			
		bBLm/TRn			
		&bBLm/TRn			
		lbBLm/TRn			
	Active step change instruction	SCHG (D)	Not provided	Provided	Appendix 2.4
	Transition control instruction	sTRn	Not provided	Provided	Appendix 2.3
		sBLm/TRn			
		rTRn			
		rBLm/TRn			
	Block switching instruction	BRSET (S)	Not provided *2	Provided	–
SFC program for program execution management			Not provided	Provided	–
Program execution type setting			Not provided *1 (Fixed to "scan execution type")	Provided	–

*1: For the Universal model QCPU and LCPU, the execution type of the program can be set.

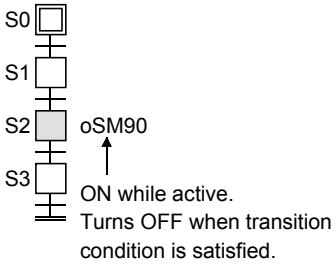
*2: Available with the Universal model QCPU whose serial number (first five digits) is "13102" or later.

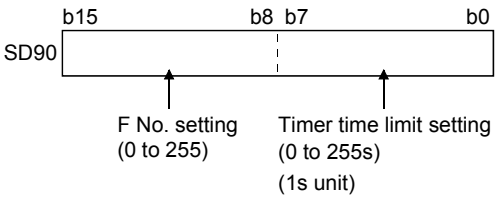
APPENDIX 2.1 Step Transition Watchdog Timer Replacement Method

(1) Operation of step transition watchdog timer

The step watchdog timer measures the ON time of the special relay for step transition watchdog timer start (SM90 to SM99), and when it exceeds the time set to the special register for step transition watchdog timer setting (SD90 to SD99), the corresponding annunciator (F) set to any of (SD90 to SD99) is turned ON.

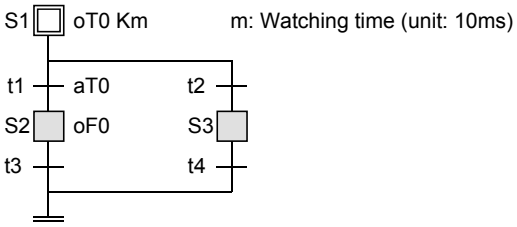
The following figure shows a step transition watchdog timer program.





(2) Step transition watchdog timer replacement method

When performing the same operation as that of the step transition watchdog timer, create the following program at the operation output.



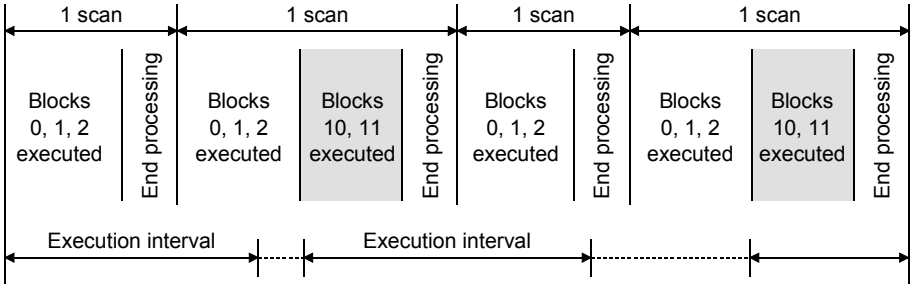
APP -8

APPENDIX 2.2 Periodic Execution Block Replacement Method

(1) Operation of periodic execution block

A periodic execution block is executed in each scan where the specified execution interval has elapsed.

The following figure shows the operation performed when blocks 0, 1, 2, 10 and 11 are used and blocks 10 and 11 are set as the periodic execution blocks.

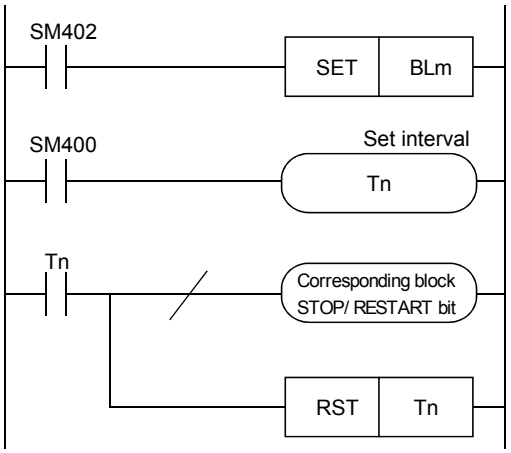


(2) Periodic execution block replacement method

When the execution interval measured by the timer in the sequence program reaches the set time, the specified block is activated by the STOP/RESTART bit.

When the set time is not reached, the block is in a stop status.

To hold the output also when the block is in a stop status, select "Change OUT instruction in specified block to SET instruction" or "Coil output held for stop-time output mode".



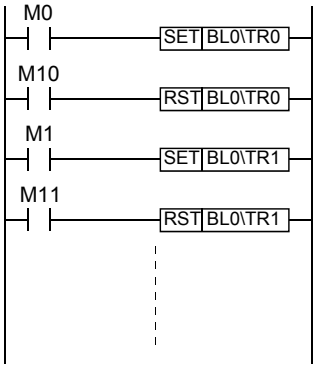
APP -9

APPENDIX 2.3 Forced Transition Bit (TRn) Replacement Method

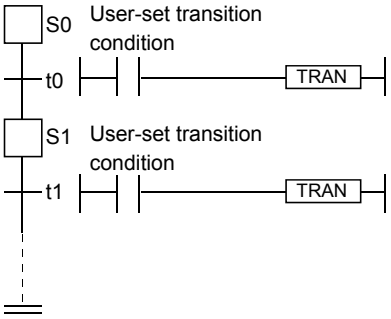
(1) Operation by forced transition bit

The forced transition bit forcibly satisfies a transition condition.
When the forced transition bits are used, the preset input conditions can be ignored and the transition conditions can be satisfied in due order.

Sequence program



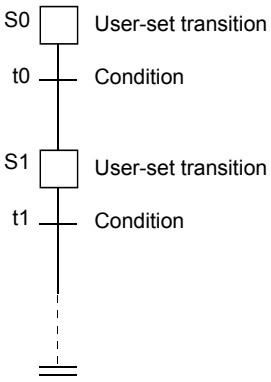
SFC program



(2) Forced transition bit replacement method

Describe any bit device in the transition condition, where it is desired to cause a forced transition, under the OR condition and turn ON the bit device described under the OR condition to cause a forced transition.

SFC program

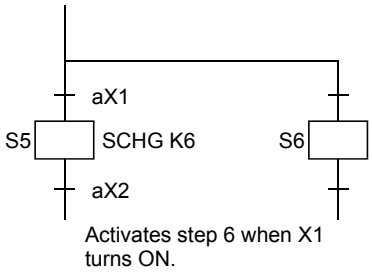


APP -10

APPENDIX 2.4 Active Step Change Instruction (SCHG) Replacement Method

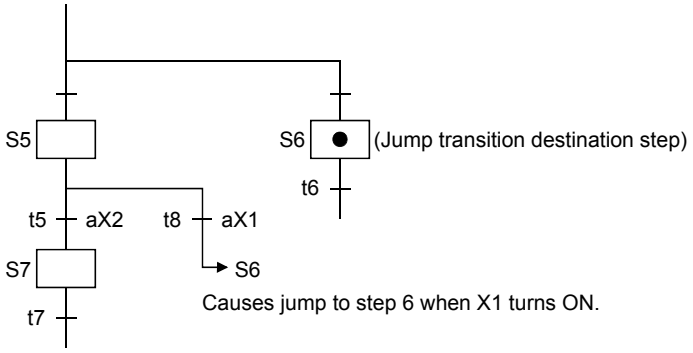
(1) Operation of active step change instruction

The active step change instruction deactivates the instruction-executed step and forcibly activates the specified step in the same block.



(2) Active step change instruction replacement method

Using a jump transition and selection branching, create a program that will cause a jump to the specified step when the transition condition is established.



APP -11

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

SH(NA)-080076-T(1406)MEE
MODEL: QCPU-P-E(SAP-L)
MODEL CODE: 13JF61

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.